



SKI.WI431.1 AT 指令集

Doc. No. : 0003-0001

Rev. : v1.3

目录

REVISION HISTORY.....	2
1. 指令集说明.....	7
2. 系统指令.....	8
2.1 AT+GET_SDK_VER - 获取客户 SDK 版本号.....	8
2.2 AT+GET_SYS_STATUS - 获取系统状态.....	8
2.3 AT+REBOOT - 重启系统.....	9
2.4 AT+LIGHT_SLEEP - Light 休眠模式.....	9
2.5 AT+MODEM_SLEEP - Modem 休眠模式.....	9
2.6 AT+DEEP_SLEEP - Deep 休眠模式.....	10
2.7 AT+ WAKEUP_GPIO - 设置唤醒休眠模式的 GPIO.....	10
2.8 AT+UART_CFG_TMP - 临时配置 UART.....	10
2.9 AT+UART_CFG - 配置 UART.....	11
2.10 AT+UART_GET_CFG - 获取 UART 配置.....	11
2.11 AT+GPIO_WRITE - GPIO 置高低.....	11
2.12 AT+GPIO_READ - 读 GPIO 电平状态.....	12
2.13 AT+GPIO_TOGGLE - 翻转 GPIO 电平状态.....	12
2.14 AT+GPIO_SET_DIR - 配置 GPIO 方向.....	12
2.15 AT+GPIO_GET_DIR - 读取 GPIO 方向配置.....	12
2.16 AT+GET_ADC - 读取 ADC.....	13
2.17 AT+SET_PWM - 设置 PWM.....	13
2.18 AT+STOP_PWM - 停止 PWM.....	13
2.19 AT+SYS_TEST - 测试系统 API.....	14
3. WIFI 指令.....	15
3.1 AT+WIFI_STATUS - 获取 WIFI 状态.....	16
3.2 AT+WSMAC - 获取当前的 mac 地址.....	16
3.3 AT+WMODE - 获取当前的模式.....	17
3.4 AT+WIFI_SET_MODE_TMP - 临时配置 STA/AP 模式.....	17
3.5 AT+WIFI_SET_MODE - 配置 STA/AP 模式.....	17
3.6 AT+WIFI_JOIN_AP_TMP - 临时配置并加入 AP.....	17
3.7 AT+WIFI_JOIN_AP - 配置并加入 AP.....	18
3.8 AT+WIFI_JOIN_AP_FAST - 配置并快速加入 AP.....	18
3.9 AT+WIFI_JOIN_AP_AUTO - 配置自动选择加密模式.....	19
3.10 AT+WIFI_DISCONNECT - 断开到 AP 的连接.....	19
3.11 AT+WIFI_SCAN - STA 扫描 AP.....	19
3.12 AT+WIFI_GET_SCANED - 获取扫描结果.....	19

3.13	AT+WIFI_AP_CFG_TMP - 临时配置并启动 AP 模式.....	20
3.14	AT+WIFI_AP_CFG - 配置并启动 AP 模式.....	20
3.15	AT+WIFI_AP_STA_LIST - 获取 AP 模式下已经连接的 station 信息.....	20
3.16	AT+SMART_CFG_START - 开始智能联网.....	20
3.17	AT+SMART_CFG_STOP - 停止智能联网.....	21
3.18	AT+WIFI_STA_MAC_TMP - 临时设置 STA MAC 地址.....	21
3.19	AT+WIFI_STA_MAC - 设置 STA MAC 地址.....	21
3.20	AT+WIFI_RESET_MAC - 恢复到出厂 MAC 地址.....	21
3.21	AT+WIFI_AP_MAC_TMP - 临时设置 AP MAC 地址.....	22
3.22	AT+WIFI_AP_MAC - 设置 AP MAC 地址.....	22
3.23	AT+WIFI_STA_IP - 设置 STA IP 地址.....	22
3.24	AT+WIFI_AP_IP - 设置 AP IP 地址.....	22
3.25	AT+WIFI_COUNTRY - 设置国家码.....	23
3.26	AT+WIFI_GET_COUNTRY - 获取国家码.....	23
3.27	AT+WIFI_CHANNEL - 设置 WIFI 信道.....	23
3.28	AT+SNTP_SET - 设置时区.....	24
3.29	AT+SNTP_GET - 获取 SNTP 时间.....	24
3.30	AT+WIFI_DHCP - 打开关闭 DHCP 功能.....	24
4.	TCP/IP 指令.....	25
4.1	AT+DNS - 域名解析功能.....	25
4.2	AT+IFCONING - 查询本地 IP 地址.....	25
4.3	AT+PING 功能.....	26
4.4	AT+SOCKETMUX - 设置多连接.....	26
4.5	AT+SOCKETSERVER - 建立 TCP 服务器.....	26
4.6	AT+SERVERMAXCONN - 设置服务器允许最大连接.....	27
4.7	AT+SOCKETMODE - 设置传输模式.....	27
4.8	AT+SOCKETCONN - 建立 TCP 连接, UDP 传输.....	27
4.9	AT+SCOCKETSEND - 发送数据.....	28
4.10	AT+SCOCKETFASTSEND - 快速发送数据.....	29
4.11	AT+SOCKETCLOSE - 关闭 TCP/UDP 传输.....	29
4.12	AT+SOCKETSTATUS - 查询网络连接状态.....	30
4.13	AT+SOCKETSTO - 修改接收超时时间.....	30
4.14	SOCKET 数据接收.....	30
4.15	AT+SSLCONN - 建立 SSL 连接.....	31
4.16	AT+SSLCLOSE - 关闭 SSL 连接.....	31
4.17	AT+SSLSEND - SSL 发送数据.....	31
4.18	SSL 接收数据.....	31
5.	MQTT 指令.....	33
5.1	AT+MQTT_CLIENT_INFO - 配置客户端设备信息.....	33
5.2	AT+MQTT_CONNECT - 建立 MQTT 连接.....	33
5.3	AT+MQTT_PUBLISH - 推送 MQTT 消息.....	34
5.4	AT+MQTT_SUBSCRIBE - 订阅 MQTT 主题.....	34

5.5	AT+MQTT_UNSUBSCRIBE - 取消 MQTT 主题订阅.....	34
5.6	AT+MQTT_CLOSE - 关闭 MQTT 连接.....	34
5.7	AT+MQTT_STATUS - 查询 MQTT 连接状态.....	35
6.	WEBSOCKET 指令.....	36
6.1	AT+WS_CONN - 向 websocket 服务端请求连接.....	36
6.2	AT+WS_DISCONN - 断开与 websocket 服务端的连接.....	36
6.3	AT+WS_SEND - 向 websocket 服务端发送数据.....	36
6.4	Websocket 接收服务器 TEXT 数据.....	37
6.5	AT+WS_SERVER - 建立一个 websocket 服务器.....	37
6.6	AT+WS_SERVER_STOP - 关闭本地 websocket 服务器.....	37
6.7	AT+WS_SERVER_LINKED - 查询本地服务器下连接的 client 信息.....	37
6.8	AT+WS_SERVER_SEND - 本地服务器向客户端发送数据.....	38
7.	测试指令.....	39
7.1	AT+WIFI ETF_START_TX - 开始射频发送测试包命令.....	39
7.2	AT+WIFI ETF_STOP_TX - 停止射频发送测试包命令.....	39
7.3	AT+WIFI ETF_START_RX - 开始射频接收测试包命令.....	39
7.4	AT+WIFI ETF_STOP_RX - 停止射频接收测试包命令.....	40
7.5	AT+WIFI ETF_RESET_RX - 清零 ETF 收包统计数据.....	40
7.6	AT+TX_POWER - 设置信号发射功率.....	40
7.7	AT+R_DCXO - 读取 DCXO 寄存器值.....	41
7.8	AT+W_DCXO - 写入 DCXO 寄存器值.....	41
7.9	AT+START_DCXO_CALI - 启动 DCXO 校准功能.....	41
7.10	AT+STOP_DCXO_CALI - 关闭 DCXO 校准功能.....	41
7.11	AT+WIFI ETF_SAVE_EFUSE - DCXO 写入 Efuse.....	42
7.12	AT+WIFI ETF_GET_EFUSE - 读取 efuse 数据.....	42
7.13	AT+DELTA_GAIN - 发射功率调整.....	42
7.14	AT+DELTA_DCXO - 设置频偏补偿随温度区间调整的列表（频偏补偿值）.....	43
7.15	AT+TEMP_THR - 设置频偏补偿随温度区间调整的列表（温度区间）.....	43
7.16	AT+ CLEAR_TEMP_DCXO - 清除上面两个 AT 命令（AT+TEMP_THR 和 AT+DELTA_DCXO）.....	43
7.17	AT+SHOW_GAIN - 显示用户设置的功率调整值和温度频率补偿列表.....	44
7.18	AT+READ_TEMP - 显示温度.....	44
7.19	AT+ADAPTIVE - 打开或关闭自适应功能.....	44
8.	上报数据.....	45
8.1	WIFI_CONNECT_SUCCESS.....	45
8.2	WIFI_CONNECT_GOT_IP.....	45
8.3	WIFI_CONNECT_LOSS.....	45
8.4	SOCKET 连接断开.....	45
8.5	SSL 连接成功.....	45
8.6	SSL 连接失败或断开连接.....	45

CONTENTS

(1) 指令集说明

ATBM6431 AT 指令集使用说明：

AT指令主要有两种应用场景

(2) 串口调试

在串口命令行终端输入AT命令，调试WIFI模组

(3) MCU无线透传功能

扩展了 MCU 的 WIFI 功能，实现无线透传功能。

AT 指令集主要分 3 部分：

(4) 系统 AT 指令： 系统相关的命令，UART, GPIO 配置等

(5) Wi-Fi AT 指令： Wi-Fi 联网命令

(6) TCP/IP AT 指令： TCP/IP 连接相关的命令

(7) RF 测试 AT 指令： 用于测试 RF 性能的测试命令

注意事项：

(8) AT 命令字段分割符是空格，可以是一个或多个空格，不能使用 tab 等字符

(9) AT+CMD 命令名内部的+号前后不能有空格

(10) 系统指令

指令	描述	指令示例
AT+GET_SDK_VER	读取客户 SDK 版本号	AT+GET_SDK_VER
AT+GET_SYS_STATUS	获取系统状态	AT+GET_SYS_STATUS
AT+REBOOT	重启系统	AT+REBOOT
AT+LIGHT_SLEEP	light 休眠模式	AT+LIGHT_SLEEP
AT+MODEM_SLEEP	modem 休眠模式	AT+MODEM_SLEEP
AT+ DEEP_SLEEP	Deep 休眠模式	AT+ DEEP_SLEEP
AT+ WAKEUP_GPIO	light 休眠模式唤醒 gpio	AT+ WAKEUP_GPIO
AT+UART_CFG_TMP	临时配置 UART	AT+UART_CFG_TMP RATE 115200 LEN 8 STOP 1 PARITY 0
AT+UART_CFG	配置 UART	AT+UART_CFG RATE 115200 LEN 8 STOP 1 PARITY 0
AT+UART_GET_CFG	获取 UART 配置	AT+UART_GET_CFG
AT+GPIO_WRITE	置GPIO 高低电平	AT+GPIO_WRITE PIN 19 VALUE 1
AT+GPIO_READ	读GPIO 电平状态	AT+GPIO_READ PIN 19
AT+GPIO_SET_DIR	配置 GPIO 方向	AT+GPIO_SET_DIR PIN 19 DIR 1
AT+GPIO_GET_DIR	读取 GPIO 方向	AT+GPIO_GET_DIR PIN 19
AT+GET_ADC	读取 ADC 数据	AT+GET_ADC
AT+SET_PWM	设置 PWM	AT+SET_PWM 2 100 100
AT+SYS_TEST	测试系统 API	AT+SYS_TEST
AT+memoryShow	获取系统 memory	AT+memoryShow (debug 使用 cmd)
AT+taskShow	获取系统 task 状态	AT+taskShow (debug 使用 cmd)

a) AT+GET_SDK_VER - 获取客户 SDK 版本号

指令	AT+GET_SDK_VER
格式	AT+GET_SDK_VER
参数	无
功能描述	读取客户 SDK 版本号
返回值	SDK_VER:1.0.5, TIME:Apr 21 2020 10:44:54
指令示例	AT+GET_SDK_VER

b) AT+GET_SYS_STATUS - 获取系统状态

指令	AT+GET_SYS_STATUS
格式	AT+GET_SYS_STATUS

参数	无
功能描述	读取当前系统状态
返回值	=IOT=RF=ATHENA_B 2GHZ Sep 5 2018 12:18:54 atbm_iot SDK version 0.2.1 flash map size 1Mbyte OS : freeRTOS wifi : 11n 1x1 20M [STA mode] mac : [00:12:12:23:23:23] status : [connect] ssid : [wifi_test_ap9] bssid : [64:09:80:69:a8:1e] channel : 6 cpu : 160Mhz memory total: 168Kbyte memory free: 87632 byte TCPIP : LWIP lwip config : lwipopts.h
指令示例	AT+GET_SYS_STATUS

c) AT+REBOOT - 重启系统

指令	AT+REBOOT
格式	AT+REBOOT
参数	无
功能描述	重启系统
返回值	无
指令示例	AT+REBOOT

d) AT+LIGHT_SLEEP - Light 休眠模式

指令	AT+LIGHT_SLEEP
格式	AT+LIGHT_SLEEP <enable>
参数	< enable >: 1: enable, 0: disable
功能描述	WIFI 模块进入 LIGHT 休眠模式
返回值	OK or ERR
指令示例	AT+LIGHT_SLEEP 1

e) AT+MODEM_SLEEP - Modem 休眠模式

指令	AT+MODEM_SLEEP
格式	AT+ MODEM _SLEEP <enable>

参数	< enable >: 1: enable, 0: disable
功能描述	WIFI 模块进入 modem 休眠模式
返回值	OK or ERR
指令示例	AT+ MODEM _SLEEP 1

f) AT+DEEP_SLEEP - Deep 休眠模式

指令	AT+DEEP_SLEEP
格式	AT+ DEEP_SLEEP <time>
参数	< time >: 唤醒时间, 单位秒; time 的设置范围为 1 ~ (24*60*60-1);
功能描述	WIFI 模块进入 deep 休眠模式
返回值	OK or ERR
指令示例	AT+ DEEP_SLEEP 3600

g) AT+ WAKEUP_GPIO - 设置唤醒休眠模式的 GPIO

指令	AT+ WAKEUP_GPIO
格式	AT+ WAKEUP_GPIO <pin> <low_level_wakeup value={1/0}>
参数	<p><pin> 0: gpio0, 1:gpio1, 2:gpio2, 3:gpio3, 20: gpio20, 21:gpio21, 22:gpio22, 23:gpio23 <i>目前支持 LIGHT SLEEP 唤醒的 gpio 为{ gpio0, gpio1, gpio2, gpio3, gpio20, gpio21, gpio22, gpio23}</i></p> <p><low_level_wakeup> 1: low level wakeup; 0: high level wakeup</p>
功能描述	用于设置 LIGHT SLEEP 情况下 GPIO 唤醒方式
返回值	OK or ERR
指令示例	AT+ WAKEUP_GPIO 2 1

h) AT+UART_CFG_TMP - 临时配置 UART

指令	AT+UART_CFG_TMP
格式	AT+UART_CFG_TMP RATE <baudrate> LEN <datalen> STOP <0 或1> PARITY <奇偶校验> RTS_CTS <flowctrl>
参数	<p><baudrate > : 波特率值, 配置与MCU 匹配的波特率</p> <p><datalen>: 数据位长度, 数据位长度可配置 5, 6, 7, 8</p> <p><stopbits>: 停止位, 1: 1bit 停止位, 2: 2bit 停止位</p> <p><parity>: 校验位, 0: None, 1: Odd, 2: Even</p> <p><flowctrl>: 流控 0: 无流控, 1: RTS_CTS 流控 2: 仅开启 CTS 流控</p>
功能描述	临时配置 UART 参数, 不保存到 flash 里面
返回值	OK or ERR
指令示例	AT+UART_CFG_TMP RATE 115200 LEN 8 STOP 1 PARITY 0 RTS_CTS 0

i) AT+UART_CFG - 配置 UART

指令	AT+UART_CFG
格式	AT+UART_CFG_TMP RATE <baudrate> LEN <datalen> STOP <0 或1> PARITY <奇偶校验> RTS_CTS <flowctrl>
参数	<baudrate> : 波特率值, 配置与MCU 匹配的波特率 <datalen>: 数据位长度, 数据位长度可配置 5, 6, 7, 8 <stopbits>: 停止位, 1: 1bit 停止位, 2: 2bit 停止位 <parity>: 校验位, 0: None, 1: Odd, 2: Even <flowctrl>: 流控, 0: 无流控, 1: RTS_CTS 流控 2: 仅开启 CTS 流控
功能描述	配置 UART 参数, 保存到 flash 里面
返回值	OK or ERR
指令示例	AT+UART_CFG RATE 115200 LEN 8 STOP 1 PARITY 0 RTS_CTS 0

j) AT+UART_GET_CFG - 获取 UART 配置

指令	AT+UART_GET_CFG
格式	AT+UART_GET_CFG
参数	无
功能描述	获取 UART 配置
返回值	<Baudrate> : 波特率值 <DataBits>: 数据位长度, 数据位长度可配置 5, 6, 7, 8 <StopBits>: 停止位, 1: 1bit 停止位, 2: 2bit 停止位 <Parity>: 校验位, 0: None, 1: Odd, 2: Even <RTS_CTS>: 流控, 0: 无流控, 1: RTS_CTS 流控 2: 仅开启 CTS 流控
指令示例	AT+UART_GET_CFG 返回值: Baudrate:115200, DataBits:8, StopBits:1, Parity:0, RTS_CTS:0 +OK

k) AT+GPIO_WRITE - GPIO 置高低

指令	AT+GPIO_WRITE
格式	AT+GPIO_WRITE PIN < pin_num> VALUE <level>
参数	< pin_num >: GPIO PIN 管脚编号 <level>: 0: 低电平 1: 高电平
功能描述	配置 GPIO 输出高低电压
返回值	OK or ERR
指令示例	AT+GPIO_SET_DIR PIN 19 DIR 1 AT+GPIO_WRITE PIN 19 VALUE 1

1) AT+GPIO_READ - 读 GPIO 电平状态

指令	AT+GPIO_READ
格式	AT+GPIO_READ PIN < pin_num >
参数	< pin_num >: GPIO PIN 管脚编号
功能描述	获取指定 GPIO PIN 的电平状态
返回值	pin:19 level:1 或者 params error!
指令示例	AT+GPIO_SET_DIR PIN 19 DIR 0 AT+GPIO_READ PIN 19

m) AT+GPIO_TOGGLE - 翻转 GPIO 电平状态

指令	AT+GPIO_TOGGLE
格式	AT+GPIO_TOGGLE < pin_num >
参数	< pin_num >: GPIO PIN 管脚编号
功能描述	翻转 GPIO 电平状态
返回值	OK 或者 params error!
指令示例	AT+GPIO_SET_DIR PIN 19 DIR 1 AT+GPIO_TOGGLE PIN 19

n) AT+GPIO_SET_DIR - 配置 GPIO 方向

指令	AT+GPIO_SET_DIR
格式	AT+GPIO_SET_DIR PIN < pin_num > DIR <pin_direction>
参数	< pin_num >: GPIO PIN 管脚编号 < pin_direction >: 0: 输入, 1: 输出
功能描述	配置指定 GPIO PIN 的输入输出方向; 0: 输入, 1: 输出
返回值	OK or ERR
指令示例	AT+GPIO_SET_DIR PIN 19 DIR 1

o) AT+GPIO_GET_DIR - 读取 GPIO 方向配置

指令	AT+GPIO_GET_DIR
格式	AT+GPIO_GET_DIR PIN < pin_num >
参数	< pin_num >: GPIO PIN 管脚编号
功能描述	获取指定 GPIO PIN 的方向状态, 0: 输入到 WIFI 芯片, 1: WIFI 芯片输出

返回值	pin:19 dir:1 或者 params error!
指令示例	AT+GPIO_GET_DIR PIN 19

p) AT+GET_ADC - 读取 ADC

指令	AT+GET_ADC
格式	AT+GET_ADC
参数	无
功能描述	输出 ADC 数据 范围为 0.0V~1.4V
返回值	ADC DATA (mV) 或者 params error!
指令示例	AT+GET_ADC

q) AT+SET_PWM - 设置 PWM

指令	AT+SET_PWM
格式	AT+SET_PWM <pin num> <PWM_Hi> <PWM_Lo>
参数	<p><pin num> : PWM Channel :</p> <p>PWM1_ch0——gpio0 PWM1_ch1 ——gpio1 PWM1_ch2 ——gpio2 PWM1_ch3——gpio3 PWM2_ch0——gpio16 PWM2_ch1 ——gpio17 PWM2_ch3 ——gpio18 PWM2_ch3——gpio19 PWM3_ch0——gpio20 PWM3_ch1 ——gpio21 PWM3_ch2 ——gpio22 PWM3_ch3——gpio23</p> <p>< PWM_Hi >: 高电平宽度</p> <p><PWM_Lo>: 低电平宽度</p>
功能描述	<p>脉冲宽度调制</p> <p>注意:</p> <p>比如要 GPIO1 实现占空比为 50%的频率为 1MHZ 的方波, 那么输入 AT+SET_PWM 1 40 40 需要注意的是 PWM_Hi 与 PWM_Lo 不能同时为 0, 如果 PWM_Hi 为 65535, PWM_Lo 为 0, 表示输出持续高电平, PWM_Hi 为 0, PWM_Lo 为 65535, 表示输出持续低电平</p>
返回值	+OK
指令示例	AT+SET_PWM 1 40 40

r) AT+STOP_PWM - 停止 PWM

指令	AT+STOP_PWM
格式	AT+STOP_PWM <pin num>
参数	<pin num> : PWM Channel :

	PWM1_ch0——gpio0 PWM1_ch1 ——gpio1 PWM1_ch2 ——gpio2 PWM1_ch3——gpio3 PWM2_ch0——gpio16 PWM2_ch1 ——gpio17 PWM2_ch3 ——gpio18 PWM2_ch3——gpio19 PWM3_ch0——gpio20 PWM3_ch1 ——gpio21 PWM3_ch2 ——gpio22 PWM3_ch3——gpio23
功能描述	脉冲宽度调制 注意： 比如要 GPIO1 实现占空比为 50%的频率为 1MHZ 的方波，那么输入 AT+SET_PWM 1 40 40 需要注意的是 PWM_Hi 与 PWM_Lo 不能同时为 0，如果 PWM_Hi 为 65535，PWM_Lo 为 0，表示输出持续高电平，PWM_Hi 为 0，PWM_Lo 为 65535，表示输出持续低电平
返回值	+OK
指令示例	AT+STOP_PWM 1

s) AT+SYS_TEST - 测试系统 API

指令	AT+SYS_TEST
格式	AT+SYS_TEST
参数	
功能描述	测试系统 API 工作状态，如果某个 API 测试失败会在 log 中打印—FAIL，如果成功会在 log 中打印—PASS
返回值	OK or ERR
指令示例	AT+SYS_TEST

(11) WIFI 指令

指令	描述	指令示例
AT+WIFI_STATUS	获取 WIFI 状态	AT+WIFI_STATUS
AT+WSMAC	获取当前的 mac 地址	AT+WSMAC
AT+WMODE	获取当前的模式	AT+WMODE
AT+WIFI_SET_MODE_TMP	临时配置 STA/AP 模式	AT+WIFI_SET_MODE_TMP STA_MODE
AT+WIFI_SET_MODE	配置 STA/AP 模式	AT+WIFI_SET_MODE STA_MODE
AT+WIFI_JOIN_AP_TMP	临时配置并加入 AP	AT+WIFI_JOIN_AP_TMP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_test_ap3"
AT+WIFI_JOIN_AP	配置并加入 AP	AT+WIFI_JOIN_AP_TMP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_test_ap3"
AT+WIFI_JOIN_AP_FAST	快速加入已知信道的AP	AT+WIFI_JOIN_AP_FAST CHANNEL 11 KEY_MGMT WPA_WPA2_PSK PWD altobeam SSID "WIFI_IOT"
AT+WIFI_JOIN_AP_AUTO	自动识别 AP 加密模式并加入	AT+WIFI_JOIN_AP_AUTO SSID <"ssid"> PWD <pwd>
AT+WIFI_DISCONNECT	断开到 AP 的连接	AT+WIFI_DISCONNECT
AT+WIFI_SCAN	STA 扫描 AP	AT+WIFI_SCAN
AT+WIFI_GET_SCANED	获取扫描结果	AT+WIFI_GET_SCANED
AT+WIFI_GET_SCANED ARRAY	扫描结果按照 RSSI 由高到低排序输出	AT+WIFI_GET_SCANED ARRAY
AT+WIFI_AP_CFG_TMP	临时配置并启动 AP 模式	AT+WIFI_AP_CFG_TMP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_testap 2"
AT+WIFI_AP_CFG	配置并启动 AP 模式	AT+WIFI_AP_CFG KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_testap 2"
AT+WIFI_AP_STA_LIST	获取 AP 模式下已经连接的 station 信息	AT+WIFI_AP_STA_LIST
AT+SMART_CFG_START	开始智能联网	AT+SMART_CFG_START
AT+SMART_CFG_STOP	停止智能联网	AT+SMART_CFG_STOP
AT+WIFI_STA_MAC_TMP	临时设置 STA MAC 地址	AT+WIFI_STA_MAC_TMP ADDR 12:34:56:78:55:aa
AT+WIFI_STA_MAC	设置 STA MAC 地址	AT+WIFI_STA_MAC ADDR 12:34:56:78:55:aa
AT+WIFI_RESET_MAC	恢复到出厂 MAC 地址	AT+WIFI_RESET_MAC
AT+WIFI_AP_MAC_TMP	临时设置 AP MAC 地址	AT+WIFI_AP_MAC_TMP ADDR 12:34:56:78:55:aa
AT+WIFI_AP_MAC	设置 AP MAC 地址	AT+WIFI_AP_MAC ADDR 12:34:56:78:55:aa
AT+WIFI_STA_IP_TMP	临时设置 STA IP 地址	AT+WIFI_STA_IP_TMP IP 192.168.1.100 NETMASK 255.255.255.0 GW 192.168.1.1
AT+WIFI_STA_IP	设置 STA IP 地址	AT+WIFI_STA_IP IP 192.168.1.100 NETMASK 255.255.255.0 GW 192.168.1.1
AT+WIFI_AP_IP_TMP	临时设置 AP IP 地址	AT+WIFI_AP_IP_TMP IP 192.168.1.100 NETMASK 255.255.255.0 GW 192.168.1.1
AT+WIFI_AP_IP	设置 AP IP 地址	AT+WIFI_AP_IP IP 192.168.1.100 NETMASK 255.255.255.0 GW 192.168.1.1

AT+WIFI_COUNTRY_TMP	临时设置国家码	AT+WIFI_COUNTRY_TMP CODE CN
AT+WIFI_COUNTRY	设置国家码	AT+WIFI_COUNTRY CODE CN
AT+WIFI_GET_COUNTRY	获取国家码	AT+WIFI_GET_COUNTRY
AT+WIFI_CHANNEL_TMP	临时设置WIFI 信道	AT+WIFI_CHANNEL_TMP NUM 6
AT+WIFI_CHANNEL	设置WIFI 信道	AT+WIFI_CHANNEL NUM 6
AT+SNTP_SET	设置时区	AT+SNTP_SET ZONE 8
AT+SNTP_GET	获取 SNTP 时间	AT+SNTP_GET
AT+WIFI_DHCP_TMP	临时打开关闭 dhcp 功能	AT+WIFI_DHCP_TMP 1
AT+WIFI_DHCP	打开关闭 dhcp 功能	AT+WIFI_DHCP 1
AT+WIFI ETF_START_TX	开始 ETF 发送测试	AT+WIFI ETF_START_TX CH 7 RATE 65
AT+WIFI ETF_STOP_TX	停止 ETF 发送测试	AT+WIFI ETF_STOP_TX
AT+WIFI ETF_START_RX	开始 ETF 接收测试	AT+WIFI ETF_START_RX CH 7
AT+WIFI ETF_STOP_RX	停止 ETF 接收测试	AT+WIFI ETF_STOP_RX

a) AT+WIFI_STATUS - 获取 WIFI 状态

指令	AT+WIFI_STATUS
格式	AT+WIFI_STATUS
参数	无
功能描述	获取 WIFI 状态
返回值	[AP mode] mac :00:11:22:33:44:59 channel :1 或者 [STA mode] mac :[00:11:22:33:44:59] status :[not connect] ssid :[wifi_testap] bssid :[00:00:00:00:00:00] channel :1 pwd :[altobeam] encype :WPA_WPA2_PSK
指令示例	AT+WIFI_STATUS

b) AT+WSMAC - 获取当前的 mac 地址

指令	AT+WSMAC
格式	AT+WSMAC
参数	无

功能描述	获取当前的 mac 地址
返回值	+ok=00:11:22:33:44:59
指令示例	AT+WSMAC

c) AT+WMODE - 获取当前的模式

指令	AT+WMODE
格式	AT+WMODE
参数	无
功能描述	获取当前的模式，有 AP 模式和 station 模式之分
返回值	+ok=STA
指令示例	AT+WMODE

d) AT+WIFI_SET_MODE_TMP - 临时配置 STA/AP 模式

指令	AT+WIFI_SET_MODE_TMP STA_MODE
格式	AT+WIFI_SET_MODE_TMP <Mode>
参数	AP_MODE 或 STA_MODE
功能描述	临时配置 STA/AP 模式，不保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_SET_MODE_TMP STA_MODE

e) AT+WIFI_SET_MODE - 配置 STA/AP 模式

指令	AT+WIFI_SET_MODE STA_MODE
格式	AT+WIFI_SET_MODE
参数	AP_MODE 或 STA_MODE
功能描述	配置 STA/AP 模式，保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_SET_MODE STA_MODE

f) AT+WIFI_JOIN_AP_TMP - 临时配置并加入 AP

指令	AT+WIFI_JOIN_AP_TMP
格式	AT+WIFI_JOIN_AP_TMP KEY_MGMT < enc> PWD < pwd > SSID < “ssid” > BSSID <bssid>
参数	<p>< enc>: 加密方式 NONE, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK</p> <p>< pwd >: 字符串参数，密码长度范围：8 ~ 64 字节 ASCII</p> <p>< “ssid” >:字符串参数，接入点名称，请注意”为英文字符，如果输入中文会将“ “ 也作为 ssid 的一部分</p> <p><bssid>:字符串参数，对于多个 ssid 一样的接入点，只连接具有本参数指定 bssid 的接入点。本参</p>

	数可以省略，省略本参数表示不指定 bssid，连接最优信号的接入点
功能描述	临时配置并加入 AP
返回值	OK or ERR
指令示例	AT+WIFI_JOIN_AP_TMP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_test_ap3" BSSID 30:45:96:54:40:a8 AT+WIFI_JOIN_AP_TMP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_test_ap3"

g) AT+WIFI_JOIN_AP - 配置并加入 AP

指令	AT+WIFI_JOIN_AP
格式	AT+WIFI_JOIN_AP KEY_MGMT <enc> PWD <pwd> SSID <"ssid"> BSSID <bssid>
参数	<enc>: 加密方式 NONE, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK <pwd>: 字符串参数，密码长度范围：8 ~ 64 字节 ASCII <"ssid">: 字符串参数，接入点名称，请注意"为英文字符，如果输入中文会将"也作为 ssid 的一部分。如果 ssid 中没有空格或者特殊字符，没有"也可以生效 <bssid>: 字符串参数，对于多个 ssid 一样的接入点，只连接具有本参数指定 bssid 的接入点。本参数可以省略，省略本参数表示不指定 bssid，连接最优信号的接入点
功能描述	配置并加入 AP，该命令会保存到 flash 中，设备如果重启或者断网会自动读取 flash 内容，自动连接 AP
返回值	OK or ERR
指令示例	AT+WIFI_JOIN_AP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_test_ap3" BSSID 30:45:96:54:40:a8 AT+WIFI_JOIN_AP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_test_ap3"

h) AT+WIFI_JOIN_AP_FAST - 配置并快速加入 AP

指令	AT+WIFI_JOIN_AP_FAST
格式	AT+WIFI_JOIN_AP_FAST CHANNEL <chan> KEY_MGMT <ecn> PWD <pwd> SSID <"ssid">
参数	<chan>: 快速连接 AP 所在的信道 <ecn>: 加密方式 OPEN, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK <pwd>: 字符串参数，密码长度范围：8 ~ 64 字节 ASCII <"ssid">: 字符串参数，接入点名称，请注意"为英文字符，如果输入中文会将"也作为 ssid 的一部分。如果 ssid 中没有空格或者特殊字符，没有"也可以生效
功能描述	配置并快速加入 AP，该命令与 AT+WIFI_JOIN_AP 命令的区别在于 AT+WIFI_JOIN_AP 在连接之前会先 scan，而 AT+WIFI_JOIN_AP_FAST 会在信道<chan>直接进行连接（极大的减少了连接 ap 的时间，目前测试快速连接时，连接 AP 的时间<150ms(不包含 dhcp 时间)），该命令会保存到 flash 中，设备如果重启或者断网会自动读取 flash 内容，自动使用快速连接方式连接 AP，如果过程中 AP 信道发生变化则命令失效，自动变为非快速连接方式
返回值	OK or ERR
指令示例	AT+WIFI_JOIN_AP_FAST CHANNEL 11 KEY_MGMT WPA_WPA2_PSK PWD altobeam SSID "WIFI_IOT"

i) AT+WIFI_JOIN_AP_AUTO - 配置自动选择加密模式

指令	AT+ WIFI_JOIN_AP_AUTO
格式	AT+WIFI_JOIN_AP_AUTO SSID <"ssid"> PWD <pwd>
参数	<"ssid">:字符串参数，接入点名称，请注意" " 为英文字符，如果输入中文会将" " 也作为 ssid 的一部分。如果 ssid 中没有空格或者特殊字符，没有" " 也可以生效 <pwd>: 字符串参数，密码长度范围：8 ~ 64 字节 ASCII
功能描述	配置并加入 AP，会自动识别 AP 的加密模式，并根据加密模式来完成加网。
返回值	OK or ERR
指令示例	AT+WIFI_JOIN_AP_AUTO SSID "WIFI_IOT" PWD altobeam

j) AT+WIFI_DISCONNECT - 断开到 AP 的连接

指令	AT+WIFI_DISCONNECT
格式	AT+WIFI_DISCONNECT
参数	无
功能描述	断开到 AP 的连接
返回值	OK or ERR
指令示例	AT+WIFI_DISCONNECT

k) AT+WIFI_SCAN - STA 扫描 AP

指令	AT+WIFI_SCAN
格式	AT+WIFI_SCAN
参数	无
功能描述	STA 扫描 AP
返回值	OK or ERR
指令示例	AT+WIFI_SCAN

1) AT+WIFI_GET_SCANED - 获取扫描结果

指令	AT+WIFI_GET_SCANED
格式	AT+WIFI_GET_SCANED ARRAY
参数	ARRAY: 可选，增加此选项，扫描结果按照 RSSI 由高到低排序输出
功能描述	STA 扫描 AP
返回值	SSID: wifi_test_ap18 channel 1 ht[1] wpa[0] rsn[0] enc[0] SSID: caifuzhongxin channel 1 ht[0] wpa[0] rsn[1] enc[1]

指令示例	AT+WIFI_GET_SCANED AT+WIFI_GET_SCANED ARRAY

m) AT+WIFI_AP_CFG_TMP - 临时配置并启动 AP 模式

指令	AT+WIFI_AP_CFG_TMP
格式	AT+WIFI_AP_CFG_TMP KEY_MGMT < ecn> PWD < pwd > SSID <" ssid" >
参数	< ecn>: 加密方式 OPEN, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK < pwd >: 字符串参数, 密码长度范围: 8 ~ 64 字节 ASCII < "ssid" >: 字符串参数, 接入点名称
功能描述	临时配置并启动 ap 模式
返回值	OK or ERR
指令示例	AT+WIFI_AP_CFG_TMP KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_testap 2"

n) AT+WIFI_AP_CFG - 配置并启动 AP 模式

指令	AT+WIFI_AP_CFG
格式	AT+WIFI_AP_CFG KEY_MGMT < ecn> PWD < pwd > SSID <" ssid" >
参数	< ecn>: 加密方式 OPEN, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK < pwd >: 字符串参数, 密码长度范围: 8 ~ 64 字节 ASCII < "ssid" >: 字符串参数, 接入点名称
功能描述	配置并启动 ap 模式
返回值	OK or ERR
指令示例	AT+WIFI_AP_CFG KEY_MGMT WPA_WPA2_PSK PWD 1234567890 SSID "wifi_testap 2"

o) AT+WIFI_AP_STA_LIST - 获取 AP 模式下已经连接的 station 信息

指令	AT+WIFI_AP_STA_LIST
格式	AT+WIFI_AP_STA_LIST
参数	无
功能描述	获取 ap 模式下已经连接的 station 信息
返回值	No[0]. station mac 00:11:22:33:44:55 connected
指令示例	AT+WIFI_AP_STA_LIST

p) AT+SMART_CFG_START - 开始智能联网

指令	AT+SMART_CFG_START
----	--------------------

格式	AT+SMART_CFG_START
参数	无
功能描述	开始智能联网
返回值	OK or ERR
指令示例	AT+SMART_CFG_START

q) AT+SMART_CFG_STOP - 停止智能联网

指令	AT+SMART_CFG_STOP
格式	AT+SMART_CFG_STOP
参数	无
功能描述	停止智能联网
返回值	OK or ERR
指令示例	AT+SMART_CFG_STOP

r) AT+WIFI_STA_MAC_TMP - 临时设置 STA MAC 地址

指令	AT+WIFI_STA_MAC_TMP
格式	AT+WIFI_STA_MAC_TMP ADDR <mac_addr>
参数	mac_addr: mac 地址值, 格式参考示例
功能描述	临时配置STA MAC 地址, 不保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_STA_MAC_TMP ADDR 12:34:56:78:55:aa

s) AT+WIFI_STA_MAC - 设置 STA MAC 地址

指令	AT+WIFI_STA_MAC
格式	AT+WIFI_STA_MAC ADDR <mac_addr>
参数	<mac_addr>: mac 地址值, 格式参考示例
功能描述	配置STA MAC 地址, 保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_STA_MAC ADDR 12:34:56:78:55:aa

t) AT+WIFI_RESET_MAC - 恢复到出厂 MAC 地址

指令	AT+WIFI_RESET_MAC
格式	AT+WIFI_RESET_MAC
参数	无
功能描述	恢复到出厂 MAC 地址
返回值	OK or ERR

指令示例	AT+WIFI_RESET_MAC
------	-------------------

u) AT+WIFI_AP_MAC_TMP - 临时设置 AP MAC 地址

指令	AT+WIFI_AP_MAC_TMP
格式	AT+WIFI_AP_MAC_TMP ADDR <mac_addr>
参数	<mac_addr>: mac 地址值, 格式参考示例
功能描述	临时配置 AP MAC 地址, 不保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_AP_MAC_TMP ADDR 12:34:56:78:55:aa

v) AT+WIFI_AP_MAC - 设置 AP MAC 地址

指令	AT+WIFI_AP_MAC
格式	AT+WIFI_AP_MAC ADDR <mac_addr>
参数	<mac_addr>: mac 地址值, 格式参考示例
功能描述	配置 AP MAC 地址, 保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_AP_MAC ADDR 12:34:56:78:55:aa

w) AT+WIFI_STA_IP - 设置 STA IP 地址

指令	AT+WIFI_STA_IP
格式	AT+WIFI_STA_IP IP <ip_addr> NETMASK <net_mask> GW <gate_way>
参数	<ip_addr>: IPV4 IP 地址 <net_mask>: 子网掩码 <gate_way>: 网关
功能描述	配置 STA IP 地址, 保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_STA_IP IP 192.168.1.100 NETMASK 255.255.255.0 GW 192.168.1.1

x) AT+WIFI_AP_IP - 设置 AP IP 地址

指令	AT+WIFI_AP_IP
格式	AT+WIFI_ AP _IP_TMP IP <ip_addr> NETMASK <net_mask> GW <gate_way>
参数	<ip_addr>: IPV4 IP 地址 <gate_way>: 网关 <net_mask>: 子网掩码

功能描述	设置 AP IP 地址，保存到 flash 里面
返回值	OK or ERR
指令示例	AT+WIFI_ AP _IP IP 192.168.1.100 NETMASK 255.255.255.0 GW 192.168.1.1

y) AT+WIFI_COUNTRY - 设置国家码

指令	AT+WIFI_COUNTRY
格式	AT+WIFI_COUNTRY CODE <CN>
参数	Country code 配置选项： "CN", "US", "EU", "JP", "CA", "AU", "IL", "MX", "FR",
功能描述	配置国家码
返回值	OK or ERR
指令示例	AT+WIFI_COUNTRY CODE CN

z) AT+WIFI_GET_COUNTRY - 获取国家码

指令	AT+WIFI_GET_COUNTRY
格式	AT+ WIFI_GET_COUNTRY
参数	无
功能描述	获取国家码
返回值	+OK=CN
指令示例	AT+WIFI_GET_COUNTRY

aa) AT+WIFI_CHANNEL - 设置 WIFI 信道

指令	AT+WIFI_CHANNEL
格式	AT+WIFI_CHANNEL NUM <num>
参数	<num>: 信道
功能描述	设置 WIFI 信道
返回值	OK or ERR
指令示例	AT+WIFI_CHANNEL NUM 6

bb) AT+SNTP_SET - 设置时区

指令	AT+SNTP_SET
格式	AT+SNTP_SET ZONE <num>
参数	<num>:时区
功能描述	设置时区
返回值	OK or ERR
指令示例	AT+SNTP_SET ZONE 8

cc) AT+SNTP_GET - 获取 SNTP 时间

指令	AT+SNTP_GET
格式	AT+SNTP_GET
参数	
功能描述	获取 SNTP 时间
返回值	time:0 date:Thu Jan 01 00:00:00 1970
指令示例	AT+SNTP_GET

dd) AT+WIFI_DHCP - 打开关闭 DHCP 功能

指令	AT+WIFI_DHCP
格式	AT+WIFI_DHCP <num>
参数	<num>:1 打开 dhcp 功能; 0 关闭 dhcp 功能
功能描述	打开关闭 dhcp 功能, 并保存到 flash 中, 下次重启依然有效, cmd 只在 station 模式下起作用 如果是打开 dhcp 功能, 将会重新进行 dhcp 如果是关闭 dhcp 功能, ip 地址将会被设置为使用 AT+WIFI_STA_IP cmd 保存到 flash 中的 ip 地址
返回值	OK or ERR
指令示例	AT+WIFI_DHCP 0 AT+WIFI_DHCP 1

(12) TCP/IP 指令

指令	描述
AT+DNS	域名解析功能
AT+IFCONFIG	查询本地 IP 地址
AT+PING	Ping 功能
AT+SOCKETMUX	设置多连接模式
AT+SOCKETSERVER	设置 TCP 服务器
AT+SERVERMAXCONN	设置服务器允许建立的最大连接数
AT+SOCKETMODE	设置透传模式
AT+SOCKETCONN	建立 TCP 连接, UDP 传输或者 SSL 连接
AT+SCOCKETSEND	发送数据
AT+SOCKETCLOSE	关闭连接
AT+SOCKETSTATUS	查询网络连接状态
AT+SSLCONN	建立 SSL 连接
AT+SSLCLOSE	关闭 SSL 连接
AT+SSLSEND	SSL 发送数据
AT+SCOCKETFASTSEND	快速发送数据 (1.0.4 版本新增指令)

a) AT+DNS - 域名解析功能

指令	AT+DNS =<IP> 功能: DNS 功能。
返回值	+ok=ipaddr:<IP address> 或者 DNS Fail +ERR
参数说明	<domain name>: 待解析的域名, 可支持长度小于 64 的域名
示例	AT+DNS = www.baidu.com

b) AT+IFCONING - 查询本地 IP 地址

指令	AT+IFCONFIG
返回值	ip: 192.168.1.117, netmask: 255.255.255.0, gw: 192.168.1.1
示例	AT+IFCONFIG

c) AT+PING 功能

指令	AT+PING=<IP> 功能：ping 功能。
返回值	+<time> OK 或 ping timeout ping completed
参数说明	<ul style="list-style-type: none">• <IP>: 字符串参数, IP 地址• <time>: ping 响应时间
示例	AT+PING=192.168.1.1

d) AT+SOCKETMUX - 设置多连接

指令	AT+SOCKETMUX =<mode> 功能：设置多连接。
返回值	+OK
参数说明	<mode>: <ul style="list-style-type: none">▸ 0: 单连接模式▸ 1: 多连接模式▸ 2: 查询当前模式
注意	<ul style="list-style-type: none">• 默认为单连接;• 只有非透传模式 (AT+ SOCKETMODE =0), 才能设置为多连接;• 必须在没有连接建立的情况下, 设置连接模式;• 如果建立了TCP 服务器, 想切换为单连接, 必须关闭服务器 (AT+SOCKETSERVER=0), 服务器仅支持多连接。
示例	AT+SOCKETMUX=1

e) AT+SOCKETSERVER - 建立 TCP 服务器

指令	AT+SOCKETSERVER=<mode> <port>
返回值	+OK
参数说明	<mode>: <ul style="list-style-type: none">▸ 0: 关闭服务器▸ 1: 建立服务器 <ul style="list-style-type: none">• <port>: 端口口号。
注意	<ul style="list-style-type: none">• 多连接情况下 (AT+SOCKETMUX=1), 才能开启 TCP 服务器。• 创建 TCP 服务器后, 自动建立 TCP 服务器监听。• 当有 TCP 客户端接入, 会自动占用一个连接 ID。

示例	AT+SOCKETMUX=1 AT+SOCKETSERVER=1,1001
----	--

f) AT+SERVERMAXCONN - 设置服务器允许最大连接

指令	查 询 指 令 : AT+SOCKETSERVERMAXCONN? 功能：查询服务器允许建立的最大连接数。	设 置 指 令 : AT+SOCKETSERVERMAXCONN=<num> 功能：设置服务器允许建立的最大连接数。
返回值	+SERVERMAXCONN:<mode> +OK	+OK
参数说明	<num>：服务器允许建立的最大连接数，取值范围：[1, 4]。	
注意	如需设置最大连接数，请在创建服务器之前设置。	
示例	AT+SOCKETMUX =1 AT+SERVERMAXCONN=2 AT+SOCKETSERVER=1,80	

g) AT+SOCKETMODE - 设置传输模式

指令	查 询 指 令 : AT+SOCKETMODE=2 功能：查询传输模式。。	设 置 指 令 : AT+SOCKETMODE=<mode> 功能：设置传输模式。
返回值	+SOCKETMODE:<mode> +OK	+OK
参数说明	<mode>: <ul style="list-style-type: none"> 0：普通传输模式 1：透传模式，仅支持 TCP 单连接和 UDP 固定通信对端的情况 2：查询传输模式 	
注意	<ul style="list-style-type: none"> 本设置不保存到 Flash。 透传模式传输时，如果连接断开，ATBM 会不停尝试重连，此时单独输入 +++ 退出透传，则停止重连；普通传输模式则不会重连，提示连接断开。 	
示例	AT+SOCKETMODE=1	

h) AT+SOCKETCONN - 建立 TCP 连接，UDP 传输

建立 TCP 连接

指令	TCP 单连接 (AT+SOCKETMUX=0) 时: AT+SOCKETCONN = <type>, <remote IP>, <remote port>	TCP 多连接 (AT+SOCKETMUX=1) 时: AT+SOCKETCONN = <link ID>, <type>, <remote IP>, <remote port>
返回值	+OK 或 +ERR 连接已经存在返回 ALREADY CONNECTED	
参数说明	* <link ID> : 网络连接 ID(0~3), 用于多连接的情况 * <type> : 连接类型, "TCP", "UDP", "SSL" * <remote IP> : 远端 IP 地址 * <remote port>: 远端端口号	
示例	AT+SOCKETCONN =TCP, 192. 168. 1. 149, 3775 AT+SOCKETCONN =TCP, app. altobeam. cn, 3775 详细请参考附录使用示例	

建立 UDP 连接

指令	单连接 (AT+SOCKETMUX=0) 时: AT+SOCKETCONN = <type>, <remoteIP>, <remote port>, [<UDP local port>]	多连接 (AT+SOCKETMUX=1) 时: AT+SOCKETCONN = <link ID>, <type>, <remote IP>, <remote port>, [<UDP local port>]
返回值	+OK 或 +ERR 连接已经存在返回 ALREADY CONNECTED	
参数说明	* <link ID> : 网络连接 ID(0~3), 用于多连接的情况 * <type> : 连接类型, "TCP", "UDP", "SSL" * <remote IP> : 远端 IP 地址 * <remote port>: 远端端口号 * [<UDP local port>]: 设置本地端口号	
示例	AT+SOCKETCONN =UDP, 192. 168. 101. 110, 1000, 1002 详细请参考附录使用示例	

i) AT+SCOCKETSEND - 发送数据

指令	设置指令: 1. 单连接时: (+SOCKETMUX=0) AT+SCOCKETSEND=<length> 2. 多连接时: (+SOCKETMUX=1) AT+SCOCKETSEND=<link ID>, <length> 功能: 在普通传输模式时, 设置发送数据的长度。	执行指令: AT+SCOCKETSEND 功能: 在透传模式时, 开始发送数据。
----	--	--

返回值	发送指定长度的数据。 收到此命令后先换行返回 >，然后开始接收串口的输入数据，超时时间5s 当输入数据长度满 <length> 时发送数据，回到普通指令模式，等待下一条 AT 指令。 如果数据发送成功，返回： SEND OK 如果数据发送失败，返回： SEND FAIL	收到此命令后先换行返回 >。 进入透传模式发送数据，每包最大2048 字节。 当输入单独一包 +++ 时，返回普通 AT 指令模式。 本指令必须在开启透传模式以及单连接下使用。
参数说明	<ul style="list-style-type: none">• <link ID>: 网络连接 ID 号 (0 ~ 3)，用于多连接的情况• <length>: 数字参数，表明发送数据的长度，最大长度为 2048	
示例	详情请参考附录使用示例	

j) AT+SCCKETFASTSEND - 快速发送数据

指令	设置指令： 1. 单 连 接 时：(+SOCKETMUX=0) AT+SCCKETFASTSEND=<length>\r\nxxxxxxxxxxxx\r\n 2. 多连接时：(+SOCKETMUX=1) AT+SCCKETFASTSEND=<link ID>,<length>\r\nxxxxxxxx\r\n 功能：快速发送数据，指令之后紧跟需要发送的数据，如果超过 100ms 没有收到指定长度的数据则指令会超时失败 注：1.0.4 版本新增指令	
返回值	如果数据发送成功，返回： SEND OK 如果数据发送失败，返回： SEND FAIL	
参数说明	<ul style="list-style-type: none">• <link ID>: 网络连接 ID 号 (0 ~ 3)，用于多连接的情况• <length>: 数字参数，表明发送数据的长度，最大长度为 2048	
示例	AT+SCCKETFASTSEND=0,10\r\n1234567890\r\n	

k) AT+SOCKETCLOSE - 关闭 TCP/UDP 传输

指令	设置指令（用于多连接的情况）： AT+SOCKETCLOSE =<link ID>	执行指令（用于单连接的情况）： AT+SOCKETCLOSE
----	--	-----------------------------------

	功能：关闭 TCP/UDP 传输。	
返回值	多连接： Close linkID <ID> +OK 单连接： +OK	
参数说明	<link ID>：需要关闭的连接 ID 号。	-----

1) AT+SOCKETSTATUS - 查询网络连接状态

指令	AT+SOCKETSTATUS
返回值	“+SOCKETSTATUS”：<link ID>,<type>,<remote IP>,<remote port>,<mode>
参数说明	* <link ID>：网络连接 ID(0~3),用于多连接的情况 * <type>：连接类型，“TCP”，“UDP”，“SSL” * <remote IP>：远端 IP 地址 * <remote port>：远端端口号 * < mode >: ▸ 0: Wi-Fi芯片作为客户端 ▸ 1: Wi-Fi 芯片作为服务器
示例	AT+SOCKETSTATUS

m) AT+SOCKETSTO - 修改接收超时时间

指令	AT+SOCKETSTO
返回值	成功 “+OK”，失败 “+ERR”
参数说明	* <rx timeout>：tcp/ip 接收数据包的超时时间，单位为 ms；如果不修改该参数默认超时时间为 1000ms，设置的数值有效范围<100~100*1000> 代码中的用法： setsockopt(socket_s, SOL_SOCKET, SO_RCVTIMEO, &rxtimeout, sizeof(int))
示例	设置 rx timeout 为 1000ms AT+SOCKETSTO 1000

n) SOCKET 数据接收

指令	无
返回值	多连接： LinkID <0~3>, len %d:data\n RECV OK\n 单连接： data\n

	RECV OK\n
参数说明	无
示例	多连接： LinkID 0, len 9:123456789 RECV OK 单连接： 123456789 RECV OK

o) AT+SSLCONN - 建立 SSL 连接

指令	AT+SSLCONN=<host>,<remote port>
返回值	指令输入成功回复 “+OK” SSL 连接成功返回 “atbm ssl connect ok”，失败返回 “atbm ssl disconnect”
参数说明	* < host > : 对端 IP 或者 host name * < remote port > : 对端端口号
示例	AT+SSLCONN=192.168.100.101,8003

p) AT+SSLCLOSE - 关闭 SSL 连接

指令	AT+SSLCLOSE
返回值	“atbm ssl disconnect” “+OK”
参数说明	无
示例	AT+SSLCLOSE

q) AT+SSLSEND - SSL 发送数据

指令	AT+SSLSEND=len
返回值	“>” 返回此数据后可输入实际要发送的数据 “\r\n” 作为结束，或者达到数据发送的长度结束
参数说明	* len : 需要发送的数据的最大长度
示例	AT+SSLSEND=100

r) SSL 接收数据

指令	无
返回值	SSL RECV, len %d:data\n

	SSL 接收到数据会返回数据长度和实际数据
参数说明	无
示例	SSL_RECV, len 33:11111111111111111111111111111111

(13) MQTT 指令

指令	描述
AT+MQTT_CLIENT_INFO	配置 MQTT 客户端设备信息
AT+MQTT_CONNECT	启动 MQTT 连接
AT+MQTT_PUBLISH	通过已经建立的 MQTT 连接推送消息
AT+MQTT_SUBSCRIBE	向 MQTT 服务器订阅一个主题
AT+MQTT_UNSUBSCRIBE	取消已经订阅的 MQTT 主题
AT+MQTT_CLOSE	关闭 MQTT 连接
AT+MQTT_STATUS	查询当前 MQTT 的连接状态

a) AT+MQTT_CLIENT_INFO - 配置客户端设备信息

指令	AT+MQTT_CLIENT_INFO
格式	AT+MQTT_CLIENT_INFO CLIENT_ID=<ID>, USER_NAME=<user_name>, PWD=<password>
参数	CLIENT_ID: 客户端设备 ID USER_NAME: MQTT 客户端注册的用户名 PWD: 用户密码
功能描述	配置 MQTT 客户端设备信息
返回值	+OK
指令示例	AT+MQTT_CLIENT_INFO CLIENT_ID=atbm6431, USER_NAME=abcd, PWD=12345678

b) AT+MQTT_CONNECT - 建立 MQTT 连接

指令	AT+MQTT_CONNECT
格式	AT+MQTT_CONNECT= <type>, <hostname>, <host port>
参数	<type>: 连接类型, "TCP" 或 "TLS" <hostname>: MQTT 服务器地址名 <host port>: MQTT 服务端口号
功能描述	向 MQTT 服务器请求建立 MQTT 连接
返回值	+OK MQTT 连接正常, 开始提供 MQTT 通信服务 +ERR 连接错误或连接已经存在 +UNSUPPORT 配置错误, 不能正常连接
指令示例	(14) MQTT test without TLS AT+MQTT_CONNECT=TCP,iot.eclipse.org,1883 (15) MQTT test with TLS AT+MQTT_CONNECT=TLS,iot.eclipse.org,8883

a) AT+MQTT_PUBLISH - 推送 MQTT 消息

指令	AT+MQTT_PUBLISH
格式	AT+MQTT_PUBLISH= <topic>, <qos>, <message data length> 回车 <message data>
参数	<topic>: MQTT topic主题 <qos>: 服务质量等级:0-2 <message data length>: 数据内容长度, 最大长度 1024 字节
功能描述	向已经建立连接的 MQTT 服务器推送消息
返回值	+Send OK 消息推送成功 +Send ERR 消息推送失败
指令示例	AT+MQTT_PUBLISH= atbm6431/data,0,12 hello world!

b) AT+MQTT_SUBSCRIBE - 订阅 MQTT 主题

指令	AT+MQTT_SUBSCRIBE
格式	AT+MQTT_SUBSCRIBE= <topic name>, <qos>
参数	<topic>: MQTT主题 <qos>: 服务质量等级:0-2
功能描述	向MQTT 服务器订阅一个主题
返回值	+OK MQTT 订阅消息发送成功 +ERR MQTT 订阅消息发送失败
指令示例	AT+MQTT_SUBSCRIBE= atbm6431/data, 0
上报的消息	格式: +TOPIC_LEN=xx,PAYLOAD_LEN=xxx:topic, data 例如: +TOPIC_LEN=14,PAYLOAD_LEN=012:/atbm6431/data, hello world!

c) AT+MQTT_UNSUBSCRIBE - 取消 MQTT 主题订阅

指令	AT+MQTT_UNSUBSCRIBE
格式	AT+MQTT_UNSUBSCRIBE= <topic name>
参数	<topic name>: MQTT topic 主题
功能描述	取消主题的订阅
返回值	+OK MQTT 取消订阅消息发送成功 +ERR MQTT 取消订阅消息发送失败
指令示例	AT+MQTT_UNSUBSCRIBE=atbm6431/data

d) AT+MQTT_CLOSE - 关闭 MQTT 连接

指令	AT+MQTT_CLOSE
----	---------------

格式	AT+MQTT_CLOSE
参数	无
功能描述	断开与服务器之间的 MQTT 连接，断开网络
返回值	+OK MQTT 连接关闭结束 +ERR MQTT 连接之前已关闭或处于未连接状态
指令示例	AT+MQTT_CLOSE

e) AT+MQTT_STATUS - 查询 MQTT 连接状态

指令	AT+MQTT_STATUS
格式	AT+MQTT_STATUS
参数	无
功能描述	获取MQTT 客户端与服务器的连接状态
返回值	+MQTT_CONNECTED MQTT 处于连接状态 +MQTT_DISCONNECTED MQTT 处于断开连接状态 +MQTT_DISCONNECTING 正在断开连接 +MQTT_STATUS_UNINIT 未初始化未连接状态
指令示例	AT+MQTT_STATUS

(16) WEBSOCKET 指令

指令	描述
AT+WS_CONN	向 websocket 服务端请求连接
AT+WS_DISCONN	断开与 websokcet 服务端的连接
AT+WS_SEND	向 websokcet 服务端发生 TEXT 格式数据
AT+WS_SERVER	在本地建立一个 websocket 服务器
AT+WS_SERVER_STOP	关闭本地 websocket 服务器
AT+WS_SERVER_LINKED	查询 websocket 服务器上面连接的客户端信息
AT+WS_SERVER_SEND	本地服务器向指定 websokcet 客户端发送数据

a) AT+WS_CONN - 向 websocket 服务端请求连接

指令	AT+WS_CONN
格式	AT+WS_CONN=<wsurl>
参数	<wsurl>: url 格式的服务端地址
功能描述	向 websocket 服务器的指定端口建立连接
返回值	+OK 连接服务器成功后返回: CONNECT TO SERVER
指令示例	TCP 连接: AT+WS_CONN=ws://192.168.100.100:8080/test SSL 连接: AT+WS_CONN=wss://192.168.100.100:8443/test

b) AT+WS_DISCONN - 断开与 websocket 服务端的连接

指令	AT+WS_DISCONN
格式	AT+WS_DISCONN
参数	无
功能描述	断开与 websokcet 服务端的连接
返回值	SEVER DISCONNECT +OK
指令示例	AT+WS_DISCONN

c) AT+WS_SEND - 向 websocket 服务端发送数据

指令	AT+WS_SEND
格式	AT+WS_SEND = <len>
参数	<len>: 需要发生数据的最大长度, 判断到” \r\n” 会直接发送当前数据
功能描述	向 websokcet 服务端发生 TEXT 格式数据

返回值	“>” 返回此符号之后可以发送后续数据
指令示例	AT+WS_SEND=1024 XXXXXXXXXXXXXXXXXXXXXXXXXXXX

d) Websocket 接收服务器 TEXT 数据

指令	无
格式	CLIENT_RECV len(%d):xxxxx
参数	无
功能描述	接收 websokcet 服务端发送过来的数据
返回值	无
指令示例	CLIENT_RECV len(6):111111

e) AT+WS_SERVER - 建立一个 websocket 服务器

指令	AT+WS_SERVER
格式	AT+ WS_SERVER = <type>,<port>
参数	<type>: 服务器类型 TCP 或 TLS <port>: 本地端口号
功能描述	在本地建立一个 websocket 服务器, TCP 服务器可以连接 3 个客户端 SSL 服务器只能连接 1 个客户端
返回值	+OK
指令示例	TCP 模式: AT+WS_SERVER=TCP,8080 TLS 模式: AT+WS_SERVER=TLS,8443

f) AT+WS_SERVER_STOP - 关闭本地 websocket 服务器

指令	AT+WS_SERVER_STOP
格式	AT+WS_SERVER_STOP
参数	无
功能描述	断开服务器下连接的所有客户端, 关闭服务器
返回值	+OK
指令示例	AT+WS_SERVER_STOP

g) AT+ WS_SERVER_LINKED - 查询本地服务器下连接的 client 信息

指令	AT+WS_SERVER_LINKED
格式	AT+WS_SERVER_LINKED
参数	无

功能描述	查询服务器下连接的 client 的信息，TCP 服务器最多可以连接 3 个客户端，SSL 服务器只能连接一个客户端
返回值	"client ID":0, IP: 192.168.100.100, Port: 9561 "client ID":1, IP: 192.168.100.100, Port: 9570 +OK
指令示例	AT+WS_SERVER_LINKED

h) AT+ WS_SERVER_SEND - 本地服务器向客户端发送数据

指令	AT+WS_SERVER_SEND
格式	AT+WS_SERVER_SEND=<linkID>,<len>
参数	<linkID>: 客户端连接号 0~2，可以通过查询获取 <len>: 需要发送的数据最大长度，判断到" \r\n" 会直接发送当前数据
功能描述	本地服务器向客户端发送 TEXT 格式数据
返回值	">" 返回此符号之后可以发送后续数据
指令示例	AT+WS_SERVER_SEND=0,1024 xxxxxxxxxxxx

(17) 测试指令

a) AT+WIFI ETF_START_TX - 开始射频发送测试包命令

指令	AT+WIFI ETF_START_TX
格式	AT+WIFI ETF_START_TX CH <num> RATE<rate>
参数	< num >: channel num 1~14 < rate >: 速率 1~65 中的合法速率值 发送的速率 11b: “1, 2, 5.5, 11,” 11g: “6, 9, 12, 18, 24, 36, 48, 54,” 11n: “6.5, 13, 19.5, 26, 39, 52, 58.5, 65” 注意: 11n 模式下 6.5 都表示 MCS0, 13 表示 MCS1...
功能描述	配置发送 ETF 测试包
返回值	OK or ERR
指令示例	AT+WIFI ETF_START_TX CH 7 RATE 65

b) AT+WIFI ETF_STOP_TX - 停止射频发送测试包命令

指令	AT+WIFI ETF_STOP_TX
格式	AT+WIFI ETF_STOP_TX
参数	无
功能描述	配置停止发送 ETF 测试包
返回值	OK or ERR
指令示例	AT+WIFI ETF_STOP_TX

c) AT+WIFI ETF_START_RX - 开始射频接收测试包命令

指令	AT+WIFI ETF_START_RX
格式	AT+WIFI ETF_START_RX CH <num> 40M <is_40M>
参数	< num >: 设置主信道 1~14 < is_40M >: 是否使用 40M 带宽; 1 使用, 0 不使用 40M 带宽模式下信道的中心频点的计算方法如下: if (CH < 3) 中心频点= CH + 2; else 中心频点= CH - 2

功能描述	配置接收 ETF 测试包
返回值	OK or ERR
指令示例	AT+WIFI ETF_START_RX CH 7 40M 0 或者可以简化为 AT+WIFI ETF_START_RX CH 7
执行指令	AT+WIFI ETF_START_RX CH 7

d) AT+WIFI ETF_STOP_RX - 停止射频接收测试包命令

指令	AT+WIFI ETF_STOP_RX
格式	AT+WIFI ETF_STOP_RX
参数	无
功能描述	配置停止接收 ETF 测试包
返回值	OK or ERR
指令示例	AT+WIFI ETF_STOP_RX

e) AT+WIFI ETF_RESET_RX - 清零 ETF 收包统计数据

指令	AT+WIFI ETF_RESET_RX
格式	AT+WIFI ETF_RESET_RX
参数	无
功能描述	清零 ETF 收包统计数据，重新开始计数
返回值	OK or ERR
指令示例	AT+WIFI ETF_RESET_RX

f) AT+TX_POWER - 设置信号发射功率

指令	AT+TX_POWER
格式	AT+TX_POWER <delta_tx_power>
参数	delta_tx_power: 该参数范围只能是-60~60，表示的物理意义是发射功率在 default 的功率基础上增加 (delta_tx_power/10) dB，也就是最多降低 6dB 和最多增加 6dB。注意连续设置多条该命令，发射功率没有累计作用，也就是每次设置多条命令时只有最后命令起作用。 注意：该命令调整的发射功率，在芯片重启之后会失效。
功能描述	设置发射功率
返回值	OK or ERR
指令示例	AT+TX_POWER -10

g) AT+R_DCX0 - 读取 DCX0 寄存器值

指令	AT+R_DCX0
格式	AT+R_DCX0
参数	无
功能描述	读 DCX0 寄存器值
返回值	DCX0= 寄存器值
指令示例	AT+R_DCX0

h) AT+W_DCX0 - 写入 DCX0 寄存器值

指令	AT+W_DCX0
格式	AT+W_DCX0 <value>
参数	<p>Value: DCX0 寄存器是 7bit, 因此取值范围只能是 0~127。当测试仪器显示的频偏为正时, DCX0 寄存器需要往大点调; 否则往小调。注意 DCX0 调的步长和CF0 关系并不是线性的, 每次调的时候最好别超过 10 个码字。</p> <p>注意: 该命令调整的 DCX0 值, 在芯片重启之后会失效。如果不是 ETF 测试: 当打开 DCX0 校准时 (参见下面命令 AT+ START_DCX0_CALI 描述, 默认是打开的), 使用该命令后调整 DCX0 值 还是会被 DCX0 校准所更新。ETF 测试时建议先打开发包或者收包命令, 再输入该命令。</p>
功能描述	写 DCX0 寄存器值
返回值	OK or ERR
指令示例	AT+W_DCX0 20

i) AT+ START_DCX0_CALI - 启动 DCX0 校准功能

指令	AT+ START_DCX0_CALI
格式	AT+ START_DCX0_CALI
参数	无
功能描述	<p>打开 DCX0 校准功能, DCX0 校准过程如下: 如果 flash 中有保存温度和 DCX0 表, 则 DCX0 由温度决定; 否则 DCX0 由收到 wifi 包的频偏决定 (连上 ap 后由 ap 包频偏决定)。注意, 如果是 ETF 模式, DCX0 校准是不起作用的。</p> <p>不设置该命令时, 默认就是打开 DCX0 校准功能。</p>
返回值	OK or ERR
指令示例	AT+ START_DCX0_CALI

j) AT+ STOP_DCX0_CALI - 关闭 DCX0 校准功能

指令	AT+ STOP_DCX0_CALI
格式	AT+ STOP_DCX0_CALI
参数	无
功能描述	关闭 DCX0 校准功能, 这个时候 DCX0 完全寄存器值决定, 可以通过 AT 命令来 AT+W_DCX0 来修改 DCX0

	值。不设置该命令时，默认是打开 DCXO 校准功能。
返回值	OK or ERR
指令示例	AT+ STOP_DCXO_CALI

k) AT+WIFI ETF_SAVE_EFUSE - DCXO 写入 Efuse

指令	AT+ WIFI ETF_SAVE_EFUSE
格式	AT+ WIFI ETF_SAVE_EFUSE DCXO <value>
参数	Value: DCXO 寄存器是 7bit，因此取值范围只能是 0~127
功能描述	将 dcxo 值写入 Efuse，这样以后芯片每次上电使用都是新的 Efuse 值。 注意：Efuse 的容量有限，建议使用 AT+W_DCXO 命令调整 DCXO 到合适值以后，再写入 Efuse。
返回值	OK or ERR
指令示例	AT+ WIFI ETF_SAVE_EFUSE DCXO 59

1) AT+WIFI ETF_GET_EFUSE - 读取 efuse 数据

指令	AT+WIFI ETF_GET_EFUSE
格式	AT+WIFI ETF_GET_EFUSE
参数	无
功能描述	读取Efuse 数据
返回值	{EFUSE:iswrite:0,specific:0,version:1,dcxo:59,gain1:0,gain2:2,gain3:1,tjroom:20,resttrim:9,sel:0,mac:dc:29:19:20:d6:bb} 各参数返回值的含义如下。 iswrite: 0: 没有修改过DCXO 值； 1: 修改过DCXO 值 specific: 保留位，客户不使用 version: Efuse 版本号，默认 为 1 dcxo: DCXO 频偏调整值 gain1: channel 1~4 发射增益校准 值 gain2: channel 5~9 发射增益校 准值 gain3: channel 10~14 发射增 益校准值 tjroom: 对应写入Efuse 时的芯片温度，需要公式换算 resttrim: 基准电流校准值 sel: 未使用 mac: MAC 地址
指令示例	AT+WIFI ETF_GET_EFUSE

m) AT+DELTA_GAIN - 发射功率调整

指令	AT+DELTA_GAIN
格式	AT+DELTA_GAIN < delta_dcxo0> < delta_dcxo1> < delta_dcxo2>

参数	<p>< delta_dcxo0>: 是 1~4 在四个信道（低信道）的 Delta gain 调整值</p> <p>< delta_dcxo1>: 是 5~9 这四个信道（中信道）的 Delta gain 调整值</p>
----	---

	<p><delta_dcxo2>: 是 10~14 这五个信道（高信道）的 Delta gain 调整值</p> <p>做 tpc 用的 gain 是 default gain + efuse delta gain + 本命令设置的 delta gain + 温度漂移的 gain 之和。</p> <p>本命令设置的 delta gain 每个码字代表 0.25dB，只能设置【-6: 0.25: 6】，也就是本命令码字范围只能是【-24: 1: 24】</p> <p>需要注意，这个功率调整的结果会保存在 Flash 中，所以芯片重新上电之后会使用这组值进行功率调整。</p>
功能描述	发射功率调整，设置低，中，高三个信道的 Delta gain 调整值
返回值	OK or ERR
指令示例	AT+DELTA_GAIN 10 10 10

n) AT+DELTA_DCX0 - 设置频偏补偿随温度区间调整的列表（频偏补偿值）

指令	AT+DELTA_DCX0
格式	AT+DELTA_DCX0 <delta_dcxo0> < delta_dcxo1> < delta_dcxo2>< delta_dcxo13> < delta_dcxo14>
参数	<p><delta_dcxo0> 到 <delta_dcxo14>: 是一个包含 15 个有符号 char 类型数据的序列，这个参数序列必须是一个递增(相邻数据可以相等)的序列，最小值必须大于-20，最大值必须小于 128，否则会出错。</p> <p>建议最小值<delta_dcxo0>初始为 0。</p> <p>需要注意，这个补偿的 DCX0 值，是在原有 Efuse 中的 DCX0 值基础上加的，所以一般情况下，60 度以下的时候都不需要补偿。一般情况下，温度越高，这个补偿的 DCX0 值越大。</p>
功能描述	设置频偏补偿随温度区间调整的列表中的频偏补偿值，可以设置不同温度区间，DCX0 补偿多少值
返回值	OK or ERR
指令示例	AT+DELTA_DCX0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

o) AT+TEMP_THR - 设置频偏补偿随温度区间调整的列表（温度区间）

指令	AT+TEMP_THR
格式	AT+TEMP_THR <delta_temp0> <delta_temp1> <delta_temp2><delta_temp14>
参数	<p><delta_temp0>到<delta_temp14>: 是一个包含 15 个无符号 char 类型数据的序列，数据的物理意义是温度(芯片内部传感器温度)，这个参数序列必须是一个递增(相邻数据可以相等)的序列，最小值设置至少 20，最大值设置不超过 150，否则会出错误。</p> <p>建议: <delta_temp0> 至少 30, <delta_temp14>至多 140, 该数组设置温度不要均匀等分，前面设置温度可以设置稀疏一点，后面设置温度设置密一点，因为温度越高频偏漂移越大。</p>
功能描述	设置频偏补偿随温度区间调整的列表中的温度区间，可以设置不同温度区间，DCX0 补偿多少值
返回值	OK or ERR
指令示例	AT+TEMP_THR 21 22 23 24 31 32 33 34 35 36 37 38 39 40 41

p) AT+ CLEAR_TEMP_DCX0 - 清除上面两个 AT 命令（AT+TEMP_THR 和 AT+ DELTA_DCX0）

指令	AT+CLEAR_TEMP_DCX0
格式	AT+CLEAR_TEMP_DCX0
参数	无

功能描述	如果有通过 AT 命令 AT+TEMP_THR 和 AT 命令 AT+ DELTA_DCXO 在 flash 中设置温度和 dcxo 关系，则芯片在非 ETF 工作过程中：DCXO 完全由温度决定的；否则 DCXO 是由收到wifi 包 cfo 决定的。 设置该命令后，就会清掉之前 AT 命令 AT+TEMP_THR 和命令 AT+ DELTA_DCXO 在 flash 中设置值，也就是芯片在非ETF 工作过程中，DCXO 完全由收到wifi 包cfo 决定的。
返回值	OK or ERR
指令示例	AT+CLEAR_TEMP_DCXO

q) AT+SHOW_GAIN - 显示用户设置的功率调整值和温度频率补偿列表

指令	AT+SHOW_GAIN
格式	AT+SHOW_GAIN
参数	无
功能描述	显示用户设置的功率调整值和温度频偏补偿列表 注意：该命令在 AT+DELTA_GAIN, AT+TEMP_THR, AT+DELTA_DCXO 命令执行之后再执行，才会有打印，打印的就是具体执行命令设置的数值。如果这三条命令只执行过其中一条，那么也只显示这一条命令的设置结果。
返回值	deltaGain: xx delta_dcxo_value: xx delta_temp_value: xx OK
指令示例	AT+SHOW_GAIN

r) AT+READ_TEMP - 显示温度

指令	AT+READ_TEMP
格式	AT+READ_TEMP
参数	无
功能描述	显示当前芯片内部温度传感器的温度，前面的温度频偏补偿，使用的就是这个温度。这个可以用来测试和建立温度频偏补偿列表的时候使用。
返回值	temperature: xx +OK
指令示例	AT+READ_TEMP

s) AT+ADAPTIVE - 打开或关闭自适应功能

指令	AT+ADAPTIVE
格式	AT+ADAPTIVE <on/off>
参数	<on/off>: on 表示打开；off 表示关闭
功能描述	打开或关闭自适应功能，打开自适应时候，如果低噪>=-75dbm,我们就不会主动发包；关闭自适应时候，有可能会主动发包。
返回值	OK
指令示例	AT+ADAPTIVE on

(18) 上报数据

a) WIFI_CONNECT_SUCCESS

WIFI 从断开状态到连接成功

b) WIFI_CONNECT_GOT_IP

DHCP 获取到IP 地址

c) WIFI_CONNECT_LOSS

WIFI 从连接状态断开

d) SOCKET 连接断开

与 AT+SOCKET 命令配合使用，主要用于反馈 socket 连接断开

格式: "client ID":<0~3>, ERR, disconnect

例如: "client ID":0, ERR, disconnect socket 连接 0 断开

e) SSL 连接成功

与 AT+SSLCONN 配合使用，启动 SSL 连接，连接成功后返回

格式: atbm ssl connect ok

f) SSL 连接失败或断开连接

与 AT+SSLCONN 配合使用，启动 SSL 连接，连接失败，或者已经连接的 SSL 断开格

式: atbm ssl disconnect

附录

- (19) 单连接 TCP Client
- (20) 连接 AP 并成功获取 IP 地址
- (21) AT+SOCKETCONN =TCP, 192. 168. 1. 149, 3775
Response : +OK
- (22) 发送数据
AT+SOCKETSEND=6 //set data length which will be send, such as 6 bytes
>123456 //输入数据
Response : SEND OK
- (23) 接收数据:
Recv:
123456 // data= 123456
RECV OK
- (24) 关闭连接
AT+SOCKETCLOSE
Response : +OK

注意：如果输入的字节数目超过了指定设定的长度则会响应 busy, 并丢弃超出的长度数据，只发送定指长度数据，发送完成返回 SEND OK

例如：

```
AT+SOCKETSEND=6 //需要发送的数据长度为 6
>12345678 // 输入数据：12345678，输入了长度为 8 的数据，所以 78 将被丢弃
Response : busy +SEND OK
对端只能收到 123456，因为 78 被发送端丢弃了
```

- (25) 透传
注意：透传只有在单连接模式下支持
- (26) 连接 AP 并成功获取 IP 地址
- (27) 连接服务器
AT+SOCKETCONN=TCP, 192. 168. 1. 149, 3775
Response : +OK
- (28) 使能透传模式
AT+ SOCKETMODE=1
Response : +OK
- (29) 发送数据
AT+SOCKETSEND
(输入数据) //From now on, data received from UART will be transparent transmitted
to server

-
- (30) 退出透传
输入连续的“+++” 退出透传
Response : +++a
注意：退出透传发送数据，回到正常的 AT 命令模式，TCP 连接任然是保持的也可以再次发送 AT+SOCKETSEND 指令，开始透传。
- (31) 断开 TCP 连接
AT+SOCKETCLOSE
Response: +OK
- (32) 多连接 TCP Client
- (33) 连接 AP 并成功获取 IP 地址
- (34) 使能多连接模式
AT+SOCKETMUX=1
Response : +OK
- (35) AT+SOCKETCONN =0, TCP, 192. 168. 1. 149, 3775
Response : +OK
- (36) 发送数据
AT+SOCKETSEND=0,6 //set data length which will be send to Link ID 0, such as 6 bytes
>abcdef //输入数据
Response : SEND OK
- (37) 接收数据:
Recv:
LinkID 1,6: abcdef // Link ID 1,len=6,data= abcdef
RECV OK
- (38) 关闭连接
AT+SOCKETCLOSE=0
Response : +OK
注意：如果输入 AT+SOCKETCLOSE=4 将关闭所有客户端连接
- (39) 多连接 TCP Server
ATBM6431 作为服务端，必须建立多连接，即可连接多个 TCP client
- (40) 连接 AP 并成功获取 IP 地址
- (41) 使能多连接模式
- (42) AT+SOCKETMUX=1
Response : +OK
- (43) 建立 TCP server
AT+SOCKETSERVER=1,5231 //default port = 5231
Response : +OK
注意：默认最大的客户端连接数为 4，大于 4 返回 ID ERROR
- (44) 发送数据:
// ID number of connection is defaulted to be 0.
AT+SOCKETSEND=0,4 //send 4 bytes to connection NO.0
> abcd // 数据 为 abcd
Response : SEND OK

注意：如果输入的字节数目超过了指定设定的长度则会响应 busy, 并发送数据的前几个字节发送完了返

回 SEND OK

(45) 接收数据:

linkid, 4, abcd //received 4 bytes, data= abcd

(46) 断开 TCP 连接

AT+SOCKETCLOSE

Response: CLOSED OK

(47) UDP 传输

UDP 传输不区分 server 或者 client , 由指令 AT+SOCKETCONN 建立传输关系。

固定远端的 UDP 通信

UDP 通信的远端固定, 由 “AT+SOCKETCONN “分配一个连接号给这个固定连接, 通信双方不会被其他设备代替,UDP 传输必须输入本地端口号

(48) AP 并成功获取 IP 地址

(49) 使能多连接模式

AT+SOCKETMUX=1

Response : +OK

(50) 创建 UDP 传输, 例如, 分配连接 ID 为 3

AT+SOCKETCONN =3,UDP,192.168.1.149,3775,5231

Response :3,CONNECT OK

(51) 发送数据:

AT+SOCKETSEND =3,5 //send 5 bytes to transmission NO.3

>abcde // 数据 为 abcde

Response : SEND OK

(52) 接收数据:

linkid, 5, abcde //received 5 bytes, data= abcde

(53) 断开 UDP 连接

AT+SOCKETCLOSE=3

Response: 3,CLOSED OK