

ESP32 AT 指令集 与使用示例



版本 1.1
版权 © 2017

关于本手册

本文描述 ESP32 AT 指令集功能以及使用方法，并介绍几种常见的 AT 指令使用示例。文档结构如下：

章节	标题	内容
第 1 章	前言	提供用户自定义 AT 指令和烧录地址的介绍说明。
第 2 章	指令说明	提供 AT 指令的概述。
第 3 章	基础 AT 指令	列举基本功能的 AT 指令。
第 4 章	Wi-Fi 功能 AT 指令	列举 Wi-Fi 功能的 AT 指令。
第 5 章	TCP/IP 相关 AT 指令	列举网络功能相关的 AT 指令。
第 6 章	参数保存在 NVS 区域 AT 指令列表	列举保存在 NVS 区域 AT 指令。
第 7 章	AT 指令使用示例	提供几种常见的 ESP32 AT 指令使用示例。
第 8 章	OTA 功能	介绍如何在 iot.espressif.cn 上创建自己的设备，并自定义 OTA 功能。
第 9 章	问题反馈	AT 指令相关问题的反馈途径与方式。

发布说明

日期	版本	发布说明
2017.01	V1.0	首次发布。
2017.02	V1.1	更新 5.2.3。

目录

1. 前言	1
1.1. 自定义 AT 指令	1
1.2. 烧录 AT 固件	1
2. 指令说明	3
3. 基础 AT 指令	4
3.1. 基础 AT 指令一览表	4
3.2. 基础 AT 指令描述	4
3.2.1. AT—测试 AT 启动	4
3.2.2. AT+RST—重启模块	4
3.2.3. AT+GMR—查询版本信息	5
3.2.4. AT+GSLP—进入 Deep-sleep 模式	5
3.2.5. ATE—开关回显功能	5
3.2.6. AT+RESTORE—恢复出厂设置	5
3.2.7. AT+UART—UART 配置	6
3.2.8. AT+UART_CUR—设置 UART 当前临时设置，不保存到 Flash	7
3.2.9. AT+UART_DEF—设置 UART 配置，保存到 Flash	8
3.2.10. AT+SLEEP—设置 sleep 模式	9
3.2.11. AT+SYSRAM—查询当前剩余 RAM 大小	9
4. Wi-Fi 功能 AT 指令	10
4.1. 基础 Wi-Fi 功能 AT 指令一览表	10
4.2. 基础 Wi-Fi 功能 AT 指令描述	11
4.2.1. AT+CWMODE—设置 Wi-Fi 模式 (Station/SoftAP/Station+SoftAP)	11
4.2.2. AT+CWJAP—连接 AP	12
4.2.3. AT+CWLAPOPT—设置 CWLAP 指令的属性	13

4.2.4. AT+CWLAP—扫描当前可用的 AP	14
4.2.5. AT+CWQAP—断开与 AP 的连接	14
4.2.6. AT+CWSAP—配置 ESP32 SoftAP 参数	15
4.2.7. AT+CWLIF—查询连接到 ESP32 SoftAP 的 Station 信息.....	16
4.2.8. AT+CWDHCP—设置 DHCP.....	16
4.2.9. AT+CWDHCPS—设置 ESP32 SoftAP DHCP 分配的 IP 范围，保存到 Flash.....	17
4.2.10. AT+CWAUTOCONN—上电是否自动连接 AP	17
4.2.11. AT+CIPSTAMAC—设置 ESP32 Station 接口的 MAC 地址.....	18
4.2.12. AT+CIPAPMAC—设置 ESP32 SoftAP 的 MAC 地址	18
4.2.13. AT+CIPSTA—设置 ESP32 Station 的 IP 地址	19
4.2.14. AT+CIPAP—设置 ESP32 SoftAP 的 IP 地址	19
4.2.15. AT+CWSTARTSMART—开启 SmartConfig	20
4.2.16. AT+CWSTOPSMART—停止 SmartConfig.....	20
4.2.17. AT+WPS—设置 WPS 功能	21

5. TCP/IP 相关 AT 指令22

5.1. TCP/IP 指令一览表.....	22
5.2. TCP/IP 指令描述	23
5.2.1. AT+CIPSTATUS—查询网络连接信息	23
5.2.2. AT+CIPDOMAIN—域名解析功能	23
5.2.3. AT+CIPSTART—建立 TCP 连接，UDP 传输或 SSL 连接.....	24
5.2.4. AT+CIPSEND—发送数据.....	26
5.2.5. AT+CIPSENDEX—发送数据	27
5.2.6. AT+CIPCLOSE—关闭 TCP/UDP/SSL 传输	27
5.2.7. AT+CIFSR—查询本地 IP 地址.....	28
5.2.8. AT+CIPMUX—设置多连接.....	28
5.2.9. AT+CIPSERVER—建立 TCP 服务器	29
5.2.10. AT+CIPMODE—设置传输模式	29

- 5.2.11. AT+SAVETRANSLINK—保存透传到 Flash.....30
 - 5.2.12. AT+CIPSTO—设置 TCP 服务器超时时间31
 - 5.2.13. AT+CIPSNTPCFG—设置时域和 SNTP 服务器31
 - 5.2.14. AT+CIPSNTPTIME—查询 SNTP 时间.....32
 - 5.2.15. AT+CIUPDATE—通过 Wi-Fi 升级软件.....32
 - 5.2.16. AT+CIPDINFO—接收网络数据时是否提示对端 IP 和端口32
 - 5.2.17. +IPD—接收网络数据33
- 6. 设置保存在 NVS 区域 AT 指令列表34
- 7. AT 指令使用示例35
 - 7.1. 单连接 TCP 客户端35
 - 7.2. UDP 传输36
 - 7.2.1. 固定远端的 UDP 通信37
 - 7.2.2. 远端可变的 UDP 通信38
 - 7.3. 透传.....39
 - 7.3.1. TCP 客户端单连接透传.....39
 - 7.3.2. UDP 透传41
 - 7.4. 多连接 TCP 服务器44
- 8. OTA 功能46
- 9. 问题反馈.....51



1. 前言

本文描述 ESP32 AT 指令集功能以及使用方法。

指令集主要分为：基础 AT 指令、Wi-Fi 功能 AT 指令、TCP/IP 工具箱 AT 指令等。

说明：

ESP32 AT 指令集相关代码链接：<https://github.com/espressif/esp32-at>。

1.1. 自定义 AT 指令

自定义 AT 指令命名时，使用英文字符以及 _（下划线）等合法字符。

AT 基于 ESP-IDF 编译，**esp-at** 中提供了开发者自定义 AT 指令的示例。乐鑫原本提供的 AT 指令以库文件 **libat_core.a** 的形式提供，将包含在编译生成的 AT BIN 固件中。

结构体 **at_cmd_struct** 用于定义一条指令的四种类型，**/esp-at/main/at_task.c** 中提供了实现自定义 AT 指令的示例，可参考实现。

用户如需编译客制化 AT 源代码，新增自定义的 AT 指令，通过终端进入文件夹 **esp-at** 中，配置环境变量 **IDF_PATH**，如（根据自己目录情况进行设置）：

```
export IDF_PATH=/home/genmisc/software_output/xcg/esp32_at/esp-idf
```

1.2. 烧录 AT 固件

参考 **esp-at/README.md** 进行烧录，请使用乐鑫官方烧录工具，烧录时注意选择对应的 Flash 大小。

乐鑫官方烧录工具链接：

http://espressif.com/zh-hans/support/download/other-tools?keys=&field_type_tid%5B%5D=13。

BIN	烧录地址	说明
bootloader.bin	0x1000	Boot loader 区
partitions_at.bin	0x8000	partitions 区
phy_init_data.bin	0xF000	phy_init 区，RF 参数区
esp-at.bin	0x10000	OTA1 区
blank.bin	0xFA000	Non-volatile storage (NVS) 区，用于保存 Wi-Fi，AT 等参数配置
blank.bin	0x1FD000	OTA 配置区



⚠ 注意:

执行 AT 指令时，AT 指令的 UART 交互端口切换到 GPIO16 和 GPIO17，其中 GPIO16 作为 ESP32 的接收口，GPIO17 作为 ESP32 的发送口。



2. 指令说明

AT 指令可以细分为四种类型：

类型	指令格式	描述
测试指令	AT+<x>=?	该命令用于该命令用于查询设置指令的参数以及取值范围。
查询指令	AT+<x>?	该命令用于返回参数的当前值。
设置指令	AT+<x>=<...>	该命令用于设置用户自定义的参数值。
执行指令	AT+<x>	该命令用于执行受模块内部程序控制的变参数不可变的功能。

注意：

- 不是每条 AT 指令都具备上述 4 种类型的命令。
- [] 括号内为缺省值，不必填写或者可能不显示。
- 使用双引号表示字符串数据 "string"，例如：AT+CWSAP="ESP756290","21030826",1,4
- 波特率为 115200，串口设置如图 2-1 所示。
- AT 指令必须大写，以回车换行符结尾 \r\n，如图 2-2 所示。

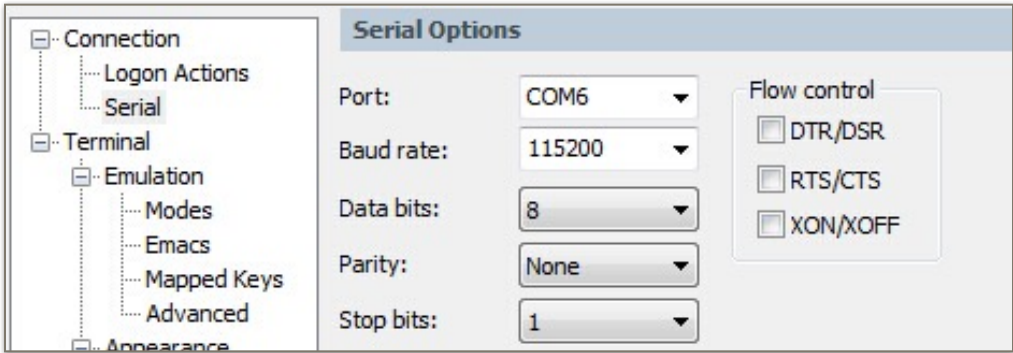


图 2-1. 串口配置

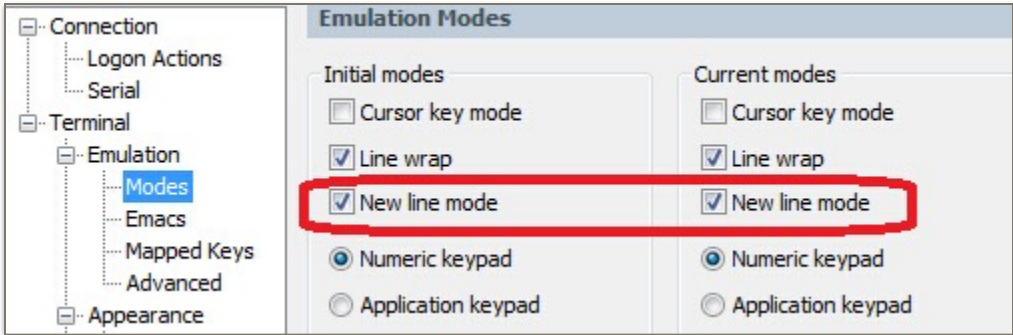


图 2-2. 新行模式



3. 基础 AT 指令

3.1. 基础 AT 指令一览表

指令	描述
AT	测试 AT 启动
AT+RST	重启模块
AT+GMR	查看版本信息
AT+GSLP	进入 Deep-sleep 模式
ATE	开关回显功能
AT+RESTORE	恢复出厂设置
AT+UART	UART 配置
AT+UART_CUR	UART 当前临时配置
AT+UART_DEF	UART 默认配置，保存到 Flash
AT+SLEEP	设置 sleep 模式
AT+SYSRAM	查询当前剩余 RAM 大小

3.2. 基础 AT 指令描述

3.2.1. AT—测试 AT 启动

执行指令	AT
响应	OK
参数说明	-

3.2.2. AT+RST—重启模块

执行指令	AT+RST
响应	OK
参数说明	-



3.2.3. AT+GMR—查询版本信息

执行指令	AT+GMR
响应	<AT version info> <SDK version info> <compile time> OK
参数说明	<ul style="list-style-type: none">• <AT version info>: AT 版本信息• <SDK version info>: SDK 版本信息• <compile time>: 编译生成时间

3.2.4. AT+GSLP—进入 Deep-sleep 模式

设置指令	AT+GSLP=<time>
响应	<time> OK
参数说明	<time>: 设置 ESP32 的睡眠时长, 单位: 毫秒。ESP32 会在休眠设定时长后自动唤醒。

3.2.5. ATE—开关回显功能

执行指令	ATE
响应	OK
参数说明	<ul style="list-style-type: none">• ATE0: 关闭回显• ATE1: 开启回显

3.2.6. AT+RESTORE—恢复出厂设置

执行指令	AT+RESTORE
响应	OK
参数说明	恢复出厂设置, 将擦除所有保存到 Flash 的参数, 恢复为默认参数。 恢复出厂设置会导致机器重启。



3.2.7. AT+UART—UART 配置

设置指令	AT+UART=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
响应	OK
参数说明	<ul style="list-style-type: none">• <baudrate>: UART 波特率• <databits>: 数据位<ul style="list-style-type: none">▸ 5: 5 bit 数据位▸ 6: 6 bit 数据位▸ 7: 7 bit 数据位▸ 8: 8 bit 数据位• <stopbits>: 停止位<ul style="list-style-type: none">▸ 1: 1 bit 停止位▸ 2: 1.5 bit 停止位▸ 3: 2 bit 停止位• <parity>: 校验位<ul style="list-style-type: none">▸ 0: None▸ 1: Odd▸ 2: Even• <flow control>: 流控<ul style="list-style-type: none">▸ 0: 不使能流控▸ 1: 使能 RTS▸ 2: 使能 CTS▸ 3: 同时使能 RTS 和 CTS
注意	<ol style="list-style-type: none">1. 本设置将保存在 NVS 区, 重新上电后仍生效。2. 使用流控需要硬件支持流控:<ul style="list-style-type: none">▸ IO15 为 UART0 CTS▸ IO14 为 UART0 RTS3. 波特率支持范围: 80 ~ 5000000
示例	AT+UART=115200,8,1,0,3



3.2.8. AT+UART_CUR—设置 UART 当前临时设置，不保存到 Flash

设置指令	AT+UART_CUR=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
响应	OK
参数说明	<ul style="list-style-type: none">• <baudrate>: UART 波特率• <databits>: 数据位<ul style="list-style-type: none">▸ 5: 5 bit 数据位▸ 6: 6 bit 数据位▸ 7: 7 bit 数据位▸ 8: 8 bit 数据位• <stopbits>: 停止位<ul style="list-style-type: none">▸ 1: 1 bit 停止位▸ 2: 1.5 bit 停止位▸ 3: 2 bit 停止位• <parity>: 校验位<ul style="list-style-type: none">▸ 0: None▸ 1: Odd▸ 2: Even• <flow control>: 流控<ul style="list-style-type: none">▸ 0: 不使能流控▸ 1: 使能 RTS▸ 2: 使能 CTS▸ 3: 同时使能 RTS 和 CTS
注意	<ol style="list-style-type: none">1. 本设置不保存在 Flash。2. 使用流控需要硬件支持流控：<ul style="list-style-type: none">▸ IO15 为 UART0 CTS▸ IO14 为 UART0 RTS3. 波特率支持范围：80 ~ 5000000。
示例	AT+UART=115200,8,1,0,3



3.2.9. AT+UART_DEF—设置 UART 配置，保存到 Flash

设置指令	AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
响应	OK
参数说明	<ul style="list-style-type: none">• <baudrate>: UART 波特率• <databits>: 数据位<ul style="list-style-type: none">▸ 5: 5 bit 数据位▸ 6: 6 bit 数据位▸ 7: 7 bit 数据位▸ 8: 8 bit 数据位• <stopbits>: 停止位<ul style="list-style-type: none">▸ 1: 1 bit 停止位▸ 2: 1.5 bit 停止位▸ 3: 2 bit 停止位• <parity>: 校验位<ul style="list-style-type: none">▸ 0: None▸ 1: Odd▸ 2: Even• <flow control>: 流控<ul style="list-style-type: none">▸ 0: 不使能流控▸ 1: 使能 RTS▸ 2: 使能 CTS▸ 3: 同时使能 RTS 和 CTS
注意	<ol style="list-style-type: none">1. 本设置将保存在 NVS 区，重新上电后仍生效。2. 使用流控需要硬件支持流控：<ul style="list-style-type: none">▸ IO15 为 UART0 CTS▸ IO14 为 UART0 RTS3. 波特率支持范围：80 ~ 5000000。
示例	AT+UART=115200,8,1,0,3



3.2.10. AT+SLEEP—设置 sleep 模式

指令	设置指令： AT+SLEEP=<sleep mode>
响应	OK 或 ERROR
参数说明	<sleep mode>： ‣ 0：禁用休眠模式 ‣ 1：Modem-sleep 模式
示例	AT+SLEEP=0

3.2.11. AT+SYSRAM—查询当前剩余 RAM 大小

查询指令	AT+SYSRAM?
响应	+SYSRAM:<remaining RAM size> OK
参数说明	<remaining RAM size>：当前剩余 RAM 大小，单位：字节
示例	AT+SYSRAM? +SYSRAM:148408 OK



4. Wi-Fi 功能 AT 指令

4.1. 基础 Wi-Fi 功能 AT 指令一览表

指令	说明
AT+CWMODE	设置 Wi-Fi 模式 (STA/AP/STA+AP)
AT+CWJAP	连接 AP
AT+CWLAPOPT	设置 AT+CWLAP 指令扫描结果的属性
AT+CWLAP	扫描附近的 AP 信息
AT+CWQAP	与 AP 断开连接
AT+CWSAP	设置 ESP32 SoftAP 配置
AT+CWLIF	获取连接到 ESP32 SoftAP 的 Station 的信息
AT+CWDHCP	设置 DHCP
AT+CWDHCPS	设置 ESP32 SoftAP DHCP 分配的 IP 范围，保存到 Flash
AT+CWAUTOCONN	设置上电时是否自动连接 AP
AT+CIPSTAMAC	设置 ESP32 Station 的 MAC 地址
AT+CIPAPMAC	设置 ESP32 SoftAP 的 MAC 地址
AT+CIPSTA	设置 ESP32 Station 的 IP 地址
AT+CIPAP	设置 ESP32 SoftAP 的 IP 地址
AT+CWSTARTSMART	开始 SmartConfig
AT+CWSTOPSMART	停止 SmartConfig
AT+WPS	设置 WPS 功能



4.2. 基础 Wi-Fi 功能 AT 指令描述

4.2.1. AT+CWMODE—设置 Wi-Fi 模式 (Station/SoftAP/Station+SoftAP)

指令	测试指令： AT+CWMODE=?	查询指令： AT+CWMODE? 功能：查询 ESP32 当前 Wi-Fi 模式。	设置指令： AT+CWMODE=<mode> 功能：设置 ESP32 当前 Wi-Fi 模式。
响应	+CWMODE:<mode> 取值列表 OK	+CWMODE:<mode> OK	OK
参数说明	<mode>： ▸ 1：Station 模式 ▸ 2：SoftAP 模式 ▸ 3：SoftAP+Station 模式		
注意	本设置将保存在 NVS 区。		
示例	AT+CWMODE=3		



4.2.2. AT+CWJAP—连接 AP

指令	查询指令： AT+CWJAP? 功能：查询 ESP32 Station 已连接的 AP 信息。	设置指令： AT+CWJAP=<ssid>,<pwd>[,<bssid>] 功能：设置 ESP32 Station 需连接的 AP。
响应	+CWJAP:<ssid>,<bssid>,<channel>,<rssi> OK	OK 或者 +CWJAP:<error code> ERROR
参数说明	<ssid>: 字符串参数，目标 AP 的 SSID	<ul style="list-style-type: none">• <ssid>: 目标 AP 的 SSID• <pwd>: 密码最长 64 字节 ASCII• [<bssid>]: 目标 AP 的 MAC 地址，一般用于有多个 SSID 相同的 AP 的情况• <error code>: (仅供参考，并不可靠)<ul style="list-style-type: none">▶ 1: 连接超时▶ 2: 密码错误▶ 3: 找不到目标 AP▶ 4: 连接失败 <p>参数设置需要开启 Station 模式，若 SSID 或者 password 中含有特殊符号，例如，或者“或者\”时，需要进行转义，其它字符转义无效。</p>
注意	本设置将保存在 NVS 区	
示例	AT+CWJAP="abc","0123456789" 例如，目标 AP 的 SSID 为 "ab\,c"，password 为 "0123456789\"，则指令如下： AT+CWJAP="ab\\,c","0123456789\\" 如果有多个 AP 的 SSID 均为 "abc"，可通过 BSSID 确定目标 AP： AT+CWJAP="abc","0123456789","ca:d7:19:d8:a6:44"	



4.2.3. AT+CWLAPOPT—设置 CWLAP 指令的属性

设置指令	AT+CWLAPOPT=<sort_enable>,<mask>
响应	OK 或 ERROR
参数说明	<ul style="list-style-type: none">• <sort_enable>: 指令 AT+CWLAP 的扫描结果是否按照信号强度 RSSI 值排序：<ul style="list-style-type: none">▸ 0: 不排序▸ 1: 为根据 RSSI 排序• <mask>: 对应 bit 若为 1, 则指令 AT+CWLAP 的扫描结果显示相关属性, 对应 bit 若为 0, 则不显示。具体如下：<ul style="list-style-type: none">▸ bit 0: 设置 AT+CWLAP 的扫描结果是否显示 <ecn>▸ bit 1: 设置 AT+CWLAP 的扫描结果是否显示 <ssid>▸ bit 2: 设置 AT+CWLAP 的扫描结果是否显示 <rssi>▸ bit 3: 设置 AT+CWLAP 的扫描结果是否显示 <mac>▸ bit 4: 设置 AT+CWLAP 的扫描结果是否显示 <ch>
示例	AT+CWLAPOPT=1,31 第一个参数为 1, 表示后续如果使用 AT+CWLAP 指令, 扫描结果将按照信号强度 RSSI 值排序; 第二个参数为 31, 即 0x1F, 表示 <mask> 的相关 bit 全部置为 1, 后续如果使用 AT+CWLAP 指令, 扫描结果将显示所有参数。



4.2.4. AT+CWLAP—扫描当前可用的 AP

指令	设置指令： AT+CWLAP=<ssid>[,<mac>,<ch>] 功能：列出符合特定条件的 AP。	执行指令： AT+CWLAP 功能：列出当前可用的 AP。
响应	+CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<ch> OK 或 ERROR	+CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<ch> OK
参数说明	<ul style="list-style-type: none">• <ecn>：加密方式<ul style="list-style-type: none">▸ 0：OPEN▸ 1：WEP▸ 2：WPA_PSK▸ 3：WPA2_PSK▸ 4：WPA_WPA2_PSK▸ 5：WPA2_Enterprise（目前 AT 不支持连接这种加密 AP）• <ssid>：字符串参数，AP 的 SSID• <rssi>：信号强度• <mac>：字符串参数，AP 的 MAC 地址	
示例	AT+CWLAP="WiFi","ca:d7:19:d8:a6:44",6 或者查找指定 SSID 的 AP： AT+CWLAP="WiFi"	

4.2.5. AT+CWQAP—断开与 AP 的连接

执行指令	AT+CWQAP
响应	OK
参数说明	-



4.2.6. AT+CWSAP—配置 ESP32 SoftAP 参数

指令	查询指令： AT+CWSAP? 功能：查询 ESP32 SoftAP 的配置参数。	设置指令： AT+CWSAP=<ssid>,<pwd>,<chl>,<ecn>[,<max conn>][,<ssid hidden>] 功能：列出当前可用的 AP。
响应	+CWSAP:<ssid>,<pwd>,<chl>,<ecn>,<max conn>,<ssid hidden>	OK 或 ERROR
参数说明	<ul style="list-style-type: none">• <ssid>：字符串参数，接入点名称• <pwd>：字符串参数，密码长度范围：8 ~ 64 字节 ASCII• <chl>：通道号• <ecn>：加密方式，不支持 WEP<ul style="list-style-type: none">▸ 0：OPEN▸ 2：WPA_PSK▸ 3：WPA2_PSK▸ 4：WPA_WPA2_PSK• [<max conn>]（选填参数）：允许连入 ESP32 SoftAP 的最多 Station 数目，取值范围 [1, 10]。• [<ssid hidden>]（选填参数）：默认为 0，开启广播 ESP32 SoftAP SSID。<ul style="list-style-type: none">▸ 0：广播 SSID▸ 1：不广播 SSID	同上。 ⚠ 注意： 指令只有在 SoftAP 模式开启后有效。
注意	本设置保存到 NVS 区域。	
示例	AT+CWSAP="ESP32","1234567890",5,3	



4.2.7. AT+CWLIF—查询连接到 ESP32 SoftAP 的 Station 信息

执行指令	AT+CWLIF
响应	<ip addr>,<mac> OK
参数说明	<ul style="list-style-type: none">• <ip addr>: 连接到 ESP32 SoftAP 的 Station IP 地址• <mac>: 连接到 ESP32 SoftAP 的 Station MAC 地址
注意	本指令无法查询静态 IP, 仅支持在 ESP32 SoftAP 和连入的 Station DHCP 均使能的情况下有效。

4.2.8. AT+CWDHCP—设置 DHCP

指令	查询指令: AT+CWDHCP?	设置指令: AT+CWDHCP=<operate>,<mode> 功能: 设置 DHCP。
响应	DHCP 是否使能	OK
参数说明	<ul style="list-style-type: none">• Bit0:<ul style="list-style-type: none">▸ 0: Station DHCP 关闭▸ 1: Station DHCP 开启• Bit1:<ul style="list-style-type: none">▸ 0: SoftAP DHCP 关闭▸ 1: SoftAP DHCP 开启	<ul style="list-style-type: none">• <operate>:<ul style="list-style-type: none">▸ 0: 关闭▸ 1: 开启• <mode>:<ul style="list-style-type: none">▸ Bit0: Station DHCP▸ Bit1: SoftAP DHCP
注意	<ul style="list-style-type: none">• 本设置保存到 NVS 区域。• 本设置指令与设置静态 IP 的指令 (AT+CIPSTA 系列和 AT+CIPAP 系列) 互相影响:<ul style="list-style-type: none">▸ 设置使能 DHCP, 则静态 IP 无效;▸ 设置静态 IP, 则 DHCP 关闭;▸ 以最后的设置为准。	
示例	AT+CWDHCP=1,1 使能 Station DHCP, 如果原 DHCP mode 为 2, 则现 DHCP mode 为 3。 AT+CWDHCP=0,2 禁能 SoftAP DHCP, 如果原 DHCP mode 为 3, 则现 DHCP mode 为 1。	



4.2.9. AT+CWDHCPS—设置 ESP32 SoftAP DHCP 分配的 IP 范围，保存到 Flash

指令	查询指令： AT+CWDHCPS?	设置指令： AT+CWDHCPS=<enable>,<lease time>,<start IP>,<end IP> 功能：设置 ESP32 SoftAP DHCP 服务器分配的 IP 范围。
响应	+CWDHCPS=<lease time>,<start IP>,<end IP>	OK
参数说明	<ul style="list-style-type: none">• <enable>:<ul style="list-style-type: none">▸ 0：清除设置 IP 范围，恢复默认值，后续参数无需填写▸ 1：使能设置 IP 范围，后续参数必须填写• <lease time>：租约时间，单位：分钟，取值范围 [1, 2880]• <start IP>：DHCP 服务器 IP 池的起始 IP• <end IP>：DHCP 服务器 IP 池的结束 IP	
注意	<ul style="list-style-type: none">• 本设置保存到 NVS 区域。• 本指令必须在 ESP32 SoftAP 模式使能，且开启 DHCP 的情况下使用，设置的 IP 范围必须与 ESP32 SoftAP 在同一网段。	
示例	AT+CWDHCPS=1,3,"192.168.4.10","192.168.4.15" 或者 AT+CWDHCPS=0 //清除设置，恢复默认值	

4.2.10. AT+CWAUTOCONN—上电是否自动连接 AP

设置指令	AT+CWAUTOCONN=<enable>
响应	OK
参数说明	<enable>: <ul style="list-style-type: none">▸ 0：上电不自动连接 AP▸ 1：上电自动连接 AP ESP32 Station 默认上电自动连接 AP。
注意	本设置保存到 NVS 区域。
示例	AT+CWAUTOCONN=1



4.2.11. AT+CIPSTAMAC—设置 ESP32 Station 接口的 MAC 地址

指令	查询指令： AT+CIPSTAMAC?	设置指令： AT+CIPSTAMAC=<mac> 功能：设置 ESP32 Station 的 MAC 地址。
响应	+CIPSTAMAC:<mac> OK	OK
参数说明	<mac>：字符串参数，ESP32 Station 的 MAC 地址	
注意	<ul style="list-style-type: none">• 本设置保存到 NVS 区域。• ESP32 SoftAP 和 Station 的 MAC 地址并不相同，请勿将其设置为同一 MAC 地址。• ESP32 MAC 地址第一个字节的 bit 0 不能为 1，例如，MAC 地址可以为 "1a:..." 但不能为 "15:..."。• FF:FF:FF:FF:FF:FF 和 00:00:00:00:00:00 为非法 MAC，无法进行设置。	
示例	AT+CIPSTAMAC="18:fe:35:98:d3:7b"	

4.2.12. AT+CIPAPMAC—设置 ESP32 SoftAP 的 MAC 地址

指令	查询指令： AT+CIPAPMAC? 功能：查询 ESP32 SoftAP 的 MAC 地址。	设置指令： AT+CIPAPMAC=<mac> 功能：设置 ESP32 SoftAP 的 MAC 地址。
响应	+CIPAPMAC:<mac> OK	OK
参数说明	<mac>：字符串参数，ESP32 SoftAP 的 MAC 地址	
注意	<ul style="list-style-type: none">• 本设置保存到 NVS 区域。• ESP32 SoftAP 和 Station 的 MAC 地址并不相同，请勿将其设置为同一 MAC 地址。• ESP32 MAC 地址第一个字节的 bit 0 不能为 1，例如，MAC 地址可以为 "18:..." 但不能为 "15:..."。• FF:FF:FF:FF:FF:FF 和 00:00:00:00:00:00 为非法 MAC，无法进行设置。	
示例	AT+CIPAPMAC="1a:fe:36:97:d5:7b"	



4.2.13. AT+CIPSTA—设置 ESP32 Station 的 IP 地址

指令	查询指令： AT+CIPSTA? 功能：查询 ESP32 Station 的 IP 地址。	设置指令： AT+CIPSTA=<ip>[,<gateway>,<netmask>] 功能：设置 ESP32 Station 的 IP 地址。
响应	+CIPSTA:<ip> OK	OK
参数说明	⚠ 注意： ESP32 Station IP 需连上 AP 后，才可以查询。	<ul style="list-style-type: none">• <ip>：字符串，ESP32 Station 的 IP 地址• [<gateway>]：网关• [<netmask>]：子网掩码
注意	<ul style="list-style-type: none">• 本设置保存到 NVS 区域。• 本设置指令与设置 DHCP 的指令（AT+CWDHCP 系列）互相影响：<ul style="list-style-type: none">▸ 设置静态 IP，则 DHCP 关闭；▸ 设置使能 DHCP，则静态 IP 无效；▸ 以最后的设置为准。	
示例	AT+CIPSTA="192.168.6.100","192.168.6.1","255.255.255.0"	

4.2.14. AT+CIPAP—设置 ESP32 SoftAP 的 IP 地址

指令	查询指令： AT+CIPAP? 功能：查询 ESP32 SoftAP 的 IP 地址。	设置指令： AT+CIPAP=<ip>[,<gateway>,<netmask>] 功能：设置 ESP32 SoftAP 的 IP 地址。
响应	+CIPAP:<ip>,<gateway>,<netmask> OK	OK
参数说明	<ul style="list-style-type: none">• <ip>：字符串，ESP32 SoftAP 的 IP 地址• [<gateway>]：网关• [<netmask>]：子网掩码	
注意	<ul style="list-style-type: none">• 本设置保存到 NVS 区域。• 目前仅支持 C 类 IP 地址。• 本设置指令与设置 DHCP 的指令（AT+CWDHCP 系列）互相影响：<ul style="list-style-type: none">▸ 设置静态 IP，则 DHCP 关闭；▸ 设置使能 DHCP，则静态 IP 无效；▸ 以最后的设置为准。	
示例	AT+CIPAP="192.168.5.1","192.168.5.1","255.255.255.0"	



4.2.15. AT+CWSTARTSMART—开启 SmartConfig

指令	执行指令： AT+CWSTARTSMART 功能：开启 SmartConfig。（SmartConfig 类型为 ESP-TOUCH+AirKiss）	设置指令： AT+CWSTARTSMART=<type> 功能：开启某指定类型的 SmartConfig。
响应	OK	
参数说明	<type>： ‣ 1: ESP-TOUCH ‣ 2: AirKiss ‣ 3: ESP-TOUCH+AirKiss	
注意	<ul style="list-style-type: none">• 用户可以参考 ESP-TOUCH 用户指南 来了解 SmartConfig 的详细介绍。• 仅支持在 ESP32 单 Station 模式下调用。• 消息 Smart get wifi info 表示 SmartConfig 成功获取到 AP 信息，之后 ESP32 尝试连接 AP，打印连接过程。• 消息 Smartconfig connected wifi 表示成功连接到 AP，此时可以调用 AT+CWSTOPSMART 停止 SmartConfig 再执行其他指令。注意，在 SmartConfig 过程中请勿执行其他指令。	
示例	AT+CWMODE=1 AT+CWSTARTSMART	

4.2.16. AT+CWSTOPSMART—停止 SmartConfig

执行指令	AT+CWSTOPSMART
响应	OK
参数说明	-
注意	无论 SmartConfig 成功与否，都请调用 AT+CWSTOPSMART 释放快连占用的内存。
示例	AT+CWSTOPSMART



4.2.17. AT+WPS—设置 WPS 功能

设置指令	AT+WPS=<enable>
响应	OK 或 ERROR
参数说明	<enable>: ▶ 1: 开启 PBC 类型的 WPS ▶ 0: 关闭 PBC 类型的 WPS
注意	<ul style="list-style-type: none">WPS 功能必须在 ESP32 Station 使能的情况下调用。WPS 不支持 WEP 加密方式。
示例	AT+CWMODE=1 AT+WPS=1



5. TCP/IP 相关 AT 指令

5.1. TCP/IP 指令一览表

指令	描述
AT+CIPSTATUS	查询网络连接信息
AT+CIPDOMAIN	域名解析功能
AT+CIPSTART	建立 TCP 连接, UDP 传输或者 SSL 连接
AT+CIPSEND	发送数据
AT+CIPSENDEX	发送数据, 达到设置长度, 或者遇到字符 \0, 则发送数据
AT+CIPCLOSE	关闭 TCP/UDP/SSL 传输
AT+CIFSR	查询本地 IP 地址
AT+CIPMUX	设置多连接模式
AT+CIPSERVER	设置 TCP 服务器
AT+CIPMODE	设置透传模式
AT+SAVETRANSLINK	保存透传连接到 Flash
AT+CIPSTO	设置 ESP32 作为 TCP 服务器时的超时时间
AT+CIUPDATE	通过 Wi-Fi 升级软件
AT+CIPDINFO	接收网络数据时, +IPD 是否提示对端 IP 和端口
AT+CIPSNTPCFG	设置时域和 SNTP 服务器
AT+CIPSNTPTIME	查询 SNTP 时间



5.2. TCP/IP 指令描述

5.2.1. AT+CIPSTATUS—查询网络连接信息

执行指令	AT+CIPSTATUS
响应	STATUS:<stat> +CIPSTATUS:<link ID>,<type>,<remote IP>,<remote port>,<local port>,<tetype>
参数说明	<ul style="list-style-type: none">• <stat>: ESP32 Station 接口的状态<ul style="list-style-type: none">▸ 2: ESP32 Station 已连接 AP, 获得 IP 地址▸ 3: ESP32 Station 已建立 TCP 或 UDP 传输▸ 4: ESP32 Station 断开网络连接▸ 5: ESP32 Station 未连接 AP• <link ID>: 网络连接 ID (0 ~ 4), 用于多连接的情况• <type>: 字符串参数, "TCP" 或者 "UDP"• <remote IP>: 字符串, 远端 IP 地址• <remote port>: 远端端口值• <local port>: ESP32 本地端口值• <tetype>:<ul style="list-style-type: none">▸ 0: ESP32 作为客户端▸ 1: ESP32 作为服务器

5.2.2. AT+CIPDOMAIN—域名解析功能

执行指令	AT+CIPDOMAIN=<domain name>
响应	+CIPDOMAIN:<IP address>
参数说明	<domain name>: 待解析的域名
示例	AT+CWMODE=1 // set Station mode AT+CWJAP="SSID","password" // access to the internet AT+CIPDOMAIN="iot.espressif.cn" // DNS function



5.2.3. AT+CIPSTART—建立 TCP 连接, UDP 传输或 SSL 连接

建立 TCP 连接

设置指令	TCP 单连接 (AT+CIPMUX=0) 时: AT+CIPSTART=<type>,<remote IP>,<remote port>[,<TCP keep alive>]	TCP 多连接 (AT+CIPMUX=1) 时: AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<TCP keep alive>]
响应	OK 或 ERROR	
参数说明	<ul style="list-style-type: none">• <link ID>: 网络连接 ID (0 ~ 4), 用于多连接的情况• <type>: 字符串参数, 连接类型, "TCP", "UDP"或"SSL"• <remote IP>: 字符串参数, 远端 IP 地址• <remote port>: 远端端口号• [<TCP keep alive>]: TCP keep-alive 侦测时间, 默认关闭此功能<ul style="list-style-type: none">▸ 0: 关闭 TCP keep-alive 功能▸ 1 ~ 7200: 侦测时间, 单位为 1s	
示例	AT+CIPSTART="TCP","iot.espressif.cn",8000 AT+CIPSTART="TCP","192.168.101.110",1000 详细请参考下文第 7 章 AT 指令使用示例 。	

建立 UDP 传输

设置指令	单连接模式 (AT+CIPMUX=0) 时: AT+CIPSTART=<type>,<remote IP>,<remote port>[,(<UDP local port>),(<UDP mode>)]	多连接模式 (AT+CIPMUX=1) 时: AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,(<UDP local port>),(<UDP mode>)]
响应	OK 或 ERROR 如果连接已经存在, 则返回: ALREADY CONNECT	



参数说明	<ul style="list-style-type: none">• <link ID>: 网络连接 ID (0 ~ 4), 用于多连接的情况• <type>: 字符串参数, 连接类型, "TCP", "UDP"或"SSL"• <remote IP>: 字符串参数, 远端 IP 地址• <remote port>: 远端端口号• [<TCP keep alive>]: TCP keep-alive 侦测时间, 默认关闭此功能<ul style="list-style-type: none">▸ 0: 关闭 TCP keep-alive 功能▸ 1 ~ 7200: 侦测时间, 单位为 1s <p>⚠ 注意:</p> <p>使用 <UDP mode> 必须先填写 <UDP local port>。</p>
示例	AT+CIPSTART="UDP","192.168.101.110",1000,1002,2 详细请参考第 7 章 AT 指令使用示例 。

建立 SSL 连接

设置指令	AT+CIPSTART=[<link ID>,<type>,<remote IP>,<remote port>[,<TCP keep alive>]
响应	OK 或 ERROR 如果连接已经存在, 则返回: ALREADY CONNECT
参数说明	<ul style="list-style-type: none">• <link ID>: 网络连接 ID (0 ~ 4), 用于多连接的情况• <type>: 字符串参数, 连接类型, "TCP", "UDP"或"SSL"• <remote IP>: 字符串参数, 远端 IP 地址• <remote port>: 远端端口号• [<TCP keep alive>]: TCP keep-alive 侦测时间, 默认关闭此功能<ul style="list-style-type: none">▸ 0: 关闭 TCP keep-alive 功能▸ 1 ~ 7200: 侦测时间, 单位为 1s
注意	<ul style="list-style-type: none">• ESP32 最多仅支持建立 1 个 SSL 连接。• SSL 连接不支持透传。• SSL 需要占用较多空间, 如果空间不足, 会导致系统重启。
示例	AT+CIPSTART="SSL","iot.espressif.cn",8443



5.2.4. AT+CIPSEND—发送数据

指令	<p>设置指令：</p> <p>1. 单连接时：(+CIPMUX=0)</p> <p>AT+CIPSEND=<length></p> <p>2. 多连接时：(+CIPMUX=1)</p> <p>AT+CIPSEND=<link ID>,<length></p> <p>3. 如果是 UDP 传输，可以设置远端 IP 和端口：</p> <p>AT+CIPSEND=[<link ID>,<length>[,<remote IP>,<remote port>]</p> <p>功能：在普通传输模式时，设置发送数据的长度。</p>	<p>执行指令：</p> <p>AT+CIPSEND</p> <p>功能：在透传模式时，开始发送数据。</p>
响应	<p>发送指定长度的数据。</p> <p>收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 length 时发送数据，回到普通指令模式，等待下一条 AT 指令。</p> <p>如果未建立连接或连接被断开，返回：</p> <p>ERROR</p> <p>如果数据发送成功，返回：</p> <p>SEND OK</p>	<p>收到此命令后先换行返回 >。</p> <p>进入透传模式发送数据，每包最大 2048 字节，或者每包数据以 20 ms 间隔区分。</p> <p>当输入单独一包 +++ 时，返回普通 AT 指令模式。发送 +++ 退出透传时，请至少间隔 1 秒再发下一条 AT 指令。</p> <p>本指令必须在开启透传模式以及单连接下使用。</p> <p>若为 UDP 透传，指令 AT+CIPSTART 参数 <UDP mode> 必须为 0。</p>
参数说明	<ul style="list-style-type: none">• <link ID>：网络连接 ID 号 (0 ~ 4)，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• [<remote IP>]：UDP 传输可以设置对端 IP• [<remote port>]：UDP 传输可以设置对端端口	-
示例	详细请参考第 7 章 AT 指令使用示例。	



5.2.5. AT+CIPSENDEX—发送数据

指令	<p>设置指令：</p> <p>1. 单连接时：(+CIPMUX=0)</p> <p>AT+CIPSENDEX=<length></p> <p>2. 多连接时：(+CIPMUX=1)</p> <p>AT+CIPSENDEX=<link ID>,<length></p> <p>3. 如果是 UDP 传输，可以设置远端 IP 和端口：</p> <p>AT+CIPSENDEX=[<link ID>,<length>[,<remote IP>,<remote port>]]</p> <p>指令功能：在普通传输模式时，设置发送数据的长度。</p>
响应	<p>发送指定长度的数据。</p> <p>收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 length 或者遇到字符 \0 时，发送数据。</p> <p>如果未建立连接或连接被断开，返回：</p> <p>ERROR</p> <p>如果数据发送成功，返回：</p> <p>SEND OK</p>
参数说明	<ul style="list-style-type: none">• <link ID>：网络连接 ID 号 (0 ~ 4)，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• 当接收数据长度满 length 或者遇到字符 \0 时，发送数据，回到普通指令模式，等待下一条 AT 指令。• 用户如需发送 \0，请转义为 \\0。

5.2.6. AT+CIPCLOSE—关闭 TCP/UDP/SSL 传输

指令	<p>设置指令（用于多连接的情况）：</p> <p>AT+CIPCLOSE=<link ID></p> <p>功能：关闭 TCP/UDP 传输。</p>	<p>执行指令（用于单连接的情况）：</p> <p>AT+CIPCLOSE</p>
响应	OK	
参数说明	<link ID>：需要关闭的连接 ID 号。当 ID 为 5 时，关闭所有连接。	-



5.2.7. AT+CIFSR—查询本地 IP 地址

执行指令	AT+CIFSR
响应	+CIFSR:<SoftAP IP address> +CIFSR:<Station IP address> OK
参数说明	<IP address>: ESP32 SoftAP 的 IP 地址 ESP32 Station 的 IP 地址
注意	ESP32 Station IP 需连上 AP 后，才可以查询。

5.2.8. AT+CIPMUX—设置多连接

指令	查询指令： AT+CIPMUX?	设置指令： AT+CIPMUX=<mode> 功能：设置连接类型。
响应	+CIPMUX:<mode> OK	OK
参数说明	<mode>： ▸ 0：单连接模式 ▸ 1：多连接模式	
注意	<ul style="list-style-type: none">默认单连接；只有非透传模式 (AT+CIPMODE=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+CIPSERVER=0)，服务器仅支持多连接。	
示例	AT+CIPMUX=1	



5.2.9. AT+CIPSERVER—建立 TCP 服务器

设置指令	AT+CIPSERVER=<mode>[,<port>]
响应	OK
参数说明	<ul style="list-style-type: none">• <mode>:<ul style="list-style-type: none">▸ 0: 关闭服务器▸ 1: 建立服务器• <port>: 端口号, 默认为 333。
注意	<ul style="list-style-type: none">• 多连接情况下 (AT+CIPMUX=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。

5.2.10. AT+CIPMODE—设置传输模式

指令	查询指令: AT+CIPMODE? 功能: 查询传输模式。	设置指令: AT+CIPMODE=<mode> 功能: 设置传输模式。
响应	+CIPMODE:<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	
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, ESP32 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+CIPMODE=1	



5.2.11. AT+SAVETRANSLINK—保存透传到 Flash

保存透传（TCP 单连接）到 Flash

设置指令	AT+SAVETRANSLINK=<mode>,<remote IP or domain name>,<remote port>[,<type>,<TCP keep alive>]
响应	OK 或 ERROR
参数说明	<ul style="list-style-type: none">• <mode>:<ul style="list-style-type: none">▸ 0: 取消开机透传▸ 1: 保存开机进入透传模式• <remote IP>: 远端 IP 或者域名• <remote port>: 远端端口• [<type>] (选填参数): TCP 或者 UDP, 缺省默认为 TCP• [<TCP keep alive>] (选填参数): TCP keep-alive 侦测, 缺省默认关闭此功能<ul style="list-style-type: none">▸ 0: 关闭 TCP keep-alive 功能▸ 1 ~ 7200: 侦测时间, 单位为秒
注意	<ul style="list-style-type: none">• 本设置将透传模式及建立的 TCP 连接均保存在 NVS 区, 下次上电自动建立 TCP 连接并进入透传。• 只要远端 IP, 端口的值符合规范, 本设置就会被保存到 Flash。
示例	AT+SAVETRANSLINK=1,"192.168.6.110",1002,"TCP"

保存透传（UDP 传输）到 Flash

设置指令	AT+SAVETRANSLINK=<mode>,<remote IP>,<remote port>,<type>[,<UDP local port>]
响应	OK 或 ERROR
参数说明	<ul style="list-style-type: none">• <mode>:<ul style="list-style-type: none">▸ 0: 取消开机透传▸ 1: 保存开机进入透传模式• <remote IP>: 远端 IP• <remote port>: 远端端口• <type>: UDP, 缺省默认为 TCP• [<UDP local port>] (选填参数): 开机进入 UDP 传输时, 使用的本地端口



注意	<ul style="list-style-type: none">• 本设置将透传模式及建立的 UDP 传输均保存在 NVS 区，下次上电自动建立 UDP 传输并进入透传。• 只要远端 IP，端口的数值符合规范，本设置就会被保存到 Flash。
示例	AT+SAVETRANSLINK=1,"192.168.6.110",1002,"UDP",1005

5.2.12. AT+CIPSTO—设置 TCP 服务器超时时间

指令	查询指令： AT+CIPSTO? 功能：查询 TCP 服务器超时时间。	设置指令： AT+CIPSTO=<time> 功能：设置 TCP 服务器超时时间。
响应	+CIPSTO:<time> OK	OK
参数说明	<time>: TCP 服务器超时时间，取值范围 0 ~ 7200s。	
注意	<ul style="list-style-type: none">• ESP32 作为 TCP 服务器，会断开一直不通信直至超时的 TCP 客户端连接。• 如果设置 AT+CIPSTO=0，则永远不会超时，不建议这样设置。	
示例	AT+CIPMUX=1 AT+CIPSERVER=1,1001 AT+CIPSTO=10	

5.2.13. AT+CIPSNTPCFG—设置时域和 SNTP 服务器

指令	查询指令： AT+CIPSNTPCFG?	执行指令： AT+CIPSNTPCFG	设置指令： AT+CIPSNTPCFG=<timezone>[,<SNTP server1>,<SNTP server2>,<SNTP server3>]
响应	+CIPSNTPCFG:<enable>,<timezone>,<SNTP server1>[,<SNTP server2>,<SNTP server3>] OK	OK	OK
参数说明	<ul style="list-style-type: none">• <enable>:<ul style="list-style-type: none">▸ 1: 配置了 SNTP 服务器信息;▸ 0: 未配置。• <timezone>: 时域，范围: [-11,13]• <SNTP server1>: 第一个 SNTP 服务器• <SNTP server2>: 第二个 SNTP 服务器• <SNTP server3>: 第三个 SNTP 服务器	清除 SNTP 服务器信息	<ul style="list-style-type: none">• <timezone>: 时域，范围: [-11,13]• <SNTP server1>: 第一个 SNTP 服务器• <SNTP server2>: 第二个 SNTP 服务器• <SNTP server3>: 第三个 SNTP 服务器 如果三个 SNTP 服务器全部省略，默认配置为: "cn.ntp.org.cn"、"ntp.sjtu.edu.cn"、"us.pool.ntp.org"。
示例	AT+CIPSNTPCFG=8,"cn.ntp.org.cn","ntp.sjtu.edu.cn"		



5.2.14. AT+CIPSNTPTIME—查询 SNTP 时间

查询指令	AT+CIPSNTPTIME?
响应	+CIPSNTPTIME:SNTP 时间 OK
参数说明	-
示例	AT+CIPSNTPCFG=8,"cn.ntp.org.cn","ntp.sjtu.edu.cn" OK AT+CIPSNTPTIME? +CIPSNTPTIME:Mon Dec 12 02:33:32 2016 OK

5.2.15. AT+CIUPDATE—通过 Wi-Fi 升级软件

执行指令	AT+CIUPDATE
响应	+CIPUPDATE:<n> OK
参数说明	<ul style="list-style-type: none">• <n>:<ul style="list-style-type: none">▸ 1: 找到服务器▸ 2: 连接到服务器▸ 3: 获得软件版本▸ 4: 开始升级
说明	<ul style="list-style-type: none">• 升级过程由于网络条件的好坏，有快慢差异。• 升级失败会提示 ERROR，请耐心等待。
注意	<ul style="list-style-type: none">• 若直接使用乐鑫提供的 AT BIN (<i>/esp-idf/bin/at</i>)，本指令将从 Espressif Cloud 下载 AT 固件升级。• 若用户自行编译 AT 源代码，则请自行实现 AT+CIUPDATE 指令的升级功能，乐鑫提供本地升级的 Demo 作为参考 (<i>/esp-idf/example/at</i>)。• 建议升级 AT 固件后，调用 AT+RESTORE 恢复出厂设置，重新初始化。

5.2.16. AT+CIPDINFO—接收网络数据时是否提示对端 IP 和端口

设置指令	AT+CIPDINFO=<mode>
响应	OK
参数说明	<mode>: <ul style="list-style-type: none">▸ 0: 不显示对端 IP 和端口▸ 1: 显示对端 IP 和端口



示例	AT+CIPDINFO=1
----	---------------

5.2.17. +IPD—接收网络数据

指令	单连接时： (+CIPMUX=0)+IPD,<len>[,<remote IP>,<remote port>]:<data>	多连接时： (+CIPMUX=1)+IPD,<link ID>,<len>[,<remote IP>,<remote port>]:<data>
参数说明	<p>此指令在普通指令模式下有效，ESP32 接收到网络数据时向串口发送 +IPD 和数据。</p> <ul style="list-style-type: none">[<remote IP>]：网络通信对端 IP，由指令 AT+CIPDINFO=1 使能显示[<remote port>]：网络通信对端端口，由指令 AT+CIPDINFO=1 使能<link ID>：收到网络连接的 ID 号<len>：数据长度<data>：收到的数据	



6. 设置保存在 NVS 区域 AT 指令列表

指令	示例
AT+UART	AT+UART=115200,8,1,0,3
AT+UART_DEF	AT+UART_DEF=115200,8,1,0,3
AT+CWDHCP	AT+CWDHCP=1,1
AT+CIPSTAMAC	AT+CIPSTAMAC="18:fe:35:98:d3:7b"
AT+CIPAPMAC	AT+CIPAPMAC="1a:fe:36:97:d5:7b"
AT+CIPSTA	AT+CIPSTA="192.168.6.100"
AT+CIPAP	AT+CIPAP="192.168.5.1"
AT+CWDHCPS	AT+CWDHCPS=1,3,"192.168.4.10","192.168.4.15"
AT+SAVETRANSLINK	AT+SAVETRANSLINK=1,"192.168.6.10",1001
AT+CWMODE	AT+CWMODE=3
AT+CWJAP	AT+CWJAP="abc","0123456789"
AT+CWSAP	AT+CWSAP="ESP32","12345678",5,3
AT+CWAUTOCONN	AT+CWAUTOCONN=1

⚠ 注意:

NVS 参数区为 0xFA000 ~ 0x110000, 88 KB。



7. AT 指令使用示例

本章介绍几种常见的 AT 指令使用示例。

7.1. 单连接 TCP 客户端

1. 设置 Wi-Fi 模式：

```
AT+CWMODE=3 // SoftAP+Station mode
```

响应：

```
OK
```

2. 连接路由：

```
AT+CWJAP="SSID","password" // SSID and password of router
```

响应：

```
OK
```

3. 查询设备 IP 地址：

```
AT+CIFSR
```

响应：

```
192.168.3.106 // device got an IP from router
```

4. PC 与 ESP32 连接同一个路由，在 PC 上使用网络调试助手，创建一个 TCP 服务器。





5. ESP32 作为客户端连接到 TCP 服务器：

```
AT+CIPSTART="TCP","192.168.3.116",8080 // protocol, server IP & port
```

6. 发送数据：

```
AT+CIPSEND=4 // set data, such as 4 bytes
>DGFY // enter the data, no CR
```

响应：

```
SEND OK
```

⚠ 注意：

若输入的字节数目超过了指令设定的长度 (n)，则会响应 `busy`，并发送数据的前 n 个字节，发送完成后响应 `SEND OK`。

7. 接收数据：

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

7.2. UDP 传输

UDP 传输不区分服务器或者客户端，由指令 `AT+CIPSTART` 建立传输关系。

1. 设置 Wi-Fi 模式：

```
AT+CWMODE=3 // SoftAP+Station mode
```

响应：

```
OK
```

2. 连接路由：

```
AT+CWJAP="SSID","password" // SSID and password of router
```

响应：

```
OK
```

3. 查询设备 IP 地址：

```
AT+CIFSR
```

响应：

```
+CIFSR:STAIP,"192.168.101.104" // IP address of ESP32 Station
```

4. PC 与 ESP32 连接同一个路由，在 PC 上使用网络调试助手，创建 UDP 传输。



下面介绍两种 UDP 通信的示例：

7.2.1. 固定远端的 UDP 通信

UDP 通信的远端固定，由 AT+CIPSTART 指令的最后参数 0 决定，分配一个连接号给这个固定连接，在通信过程中远端信息不会被改变。

1. 使能多连接：

```
AT+CIPMUX=1
```

响应：

```
OK
```

2. 创建 UDP 传输，例如，分配连接 ID 为 4。

```
AT+CIPSTART=4,"UDP","192.168.101.110",8080,1112,0
```

响应：

```
4,CONNECT
OK
```

说明：

- "192.168.101.110",8080 为 UDP 传输的远端 IP 和远端端口，也就是 PC 建立的 UDP 配置。
- 1112 为 ESP32 的 UDP 本地端口，用户可自行设置，如不设置则为随机值。
- 0 表示当前 UDP 传输建立后，UDP 远端不会被其他设备更改；即使有其他设备通过 UDP 协议发数据到 ESP32 UDP 端口 1112，ESP32 4 号 UDP 传输的远端也不会被替换，使用指令 AT+CIPSEND=4,x 发送数据，仍然是当前确定的 PC 端收到。

3. 发送数据：

```
AT+CIPSEND=4,5           // send 5 bytes to transmission NO.4
```



```
>DGFYQ // enter the data, no CR
```

响应:

```
SEND OK
```

⚠ 注意:

若输入的字节数目超过了指令设定的长度 (n), 则会响应 `busy`, 并发送数据的前 n 个字节, 发送完成后响应 `SEND OK`。

4. 接收数据:

```
+IPD,4,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

5. 断开 UDP 传输:

```
AT+CIPCLOSE=4
```

响应:

```
4,CLOSED  
OK
```

7.2.2. 远端可变的 UDP 通信

1. 创建 UDP 传输, 最后参数为 2:

```
AT+CIPSTART="UDP", "192.168.101.110", 8080, 1112, 2
```

响应:

```
CONNECT  
OK
```

📖 说明:

- "192.168.101.110", 8080 为 UDP 传输的远端 IP 和远端口, 就是前述 PC 建立的 UDP 配置。
- 1112 为 ESP32 的 UDP 本地端口, 用户可自行设置, 如不设置则为随机值。
- 2 表示当前 UDP 传输建立后, UDP 传输远端信息仍然会更改; UDP 传输的远端信息会自动更改为最近一次与 ESP32 UDP 通信的远端 IP 和端口。

2. 发送数据:

```
AT+CIPSEND=5 // send 5 bytes  
>DGFYQ // enter the data, no CR
```

响应:

```
SEND OK
```

⚠ 注意:

若输入的字节数目超过了指令设定的长度 (n), 则会响应 `busy`, 并发送数据的前 n 个字节, 发送完成后响应 `SEND OK`。



3. 若需要发 UDP 包给其他 UDP 远端，只需指定对方 IP 和端口即可。

```
AT+CIPSEND=6,"192.168.101.111",1000      // send six bytes
>abcdef                                   // enter the data, no CR
```

响应：

```
SEND OK
```

4. 接收数据：

```
+IPD,n:xxxxxxxxx                          // received n bytes, data=xxxxxxxxx
```

5. 断开 UDP 传输：

```
AT+CIPCLOSE
```

响应：

```
CLOSED
OK
```

7.3. 透传

AT Demo 仅在 ESP32 作为 TCP 客户端单连接或 UDP 传输时，支持透传。

7.3.1. TCP 客户端单连接透传

以下为 ESP32 作为 Station 实现 TCP 客户端单连接透传的举例。

1. 设置 Wi-Fi 模式：

```
AT+CWMODE=3                               // SoftAP+Station mode
```

响应：

```
OK
```

2. 连接路由：

```
AT+CWJAP="SSID","password"                // SSID and password of router
```

响应：

```
OK
```

3. 查询设备 IP 地址：

```
AT+CIFSR
```

响应：

```
192.168.101.105                           // device's IP that got from router
```

4. PC 与 ESP32 连接同一个路由，在 PC 上使用网络调试助手，创建一个 TCP 服务器。



5. 设备作为 TCP 客户端连接 TCP 服务器：

```
AT+CIPSTART="TCP","192.168.101.110",8080    // protocol, server IP & port
```

响应：

```
OK
```

6. 使能透传模式：

```
AT+CIPMODE=1
```

响应：

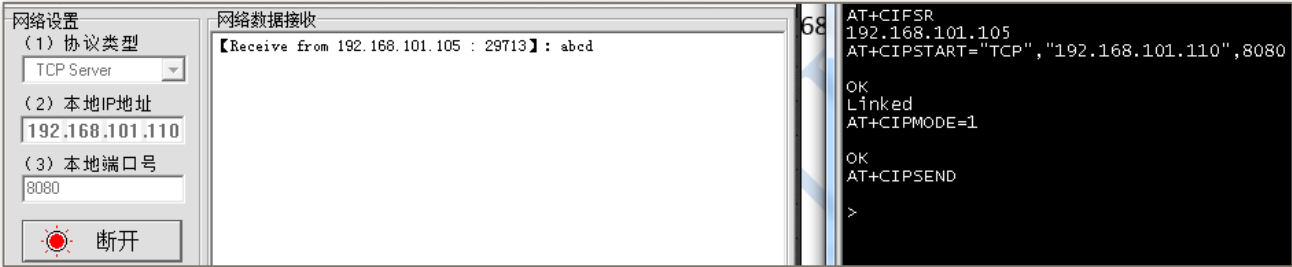
```
OK
```

7. 发送数据：

```
AT+CIPSEND
```

响应：

```
>    // From now on, data received from UART will be transparent transmitted to server
```



8. 退出发送数据：

在透传发送数据过程中，若识别到单独的一包数据 +++，则退出透传发送。请至少间隔 1 秒，再发下一条 AT 指令。

如果直接用键盘打字输入 +++，有可能时间太慢，不被认为是连续的三个 +，建议可使用如下工具：



- 字符串输入框：+++
- 发送新行：不勾选
- 点击“发送”。

⚠ 注意：

+++ 退出透传发送数据，回到正常 AT 指令模式，TCP 连接仍然是保持的，也可以再发 AT+CIPSEND 指令，开始透传。

9. 退出透传模式：

```
AT+CIPMODE=0
```

响应：

```
OK
```

10. 断开 TCP 连接：

```
AT+CIPCLOSE
```

响应：

```
CLOSED  
OK
```

7.3.2. UDP 透传

以下为 ESP32 作为 SoftAP 实现 UDP 透传的举例。



1. 设置 Wi-Fi 模式：

```
AT+CWMODE=3 // SoftAP+Station mode
```

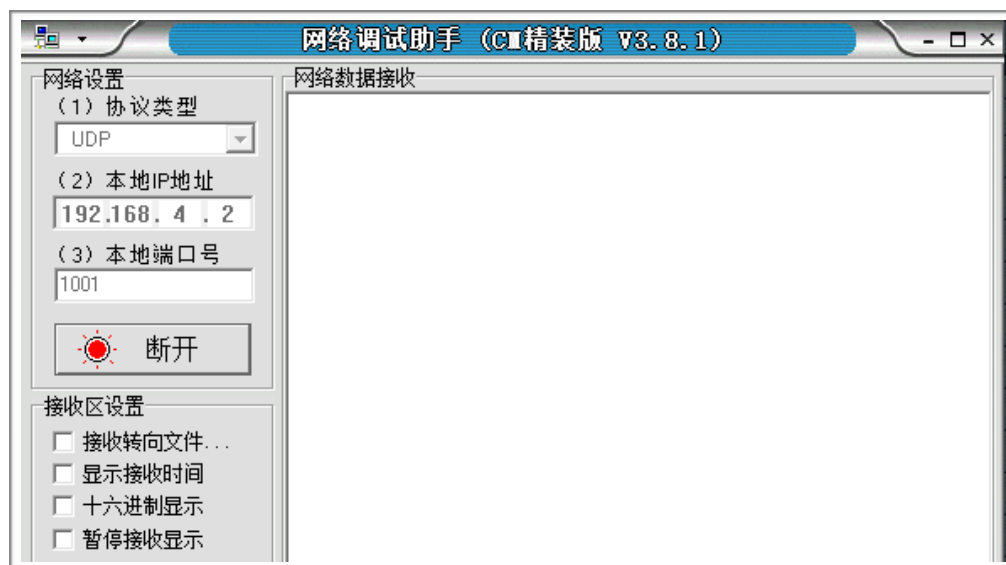
响应：

OK

2. PC 连入 ESP32 SoftAP：



3. 在 PC 上使用网络调试助手，创建一个 UDP。



4. ESP32 与 PC 对应端口建立固定对端的 UDP 传输。

```
AT+CIPSTART="UDP", "192.168.4.2", 1001, 2233, 0
```

响应：

OK

5. 使能透传模式：

```
AT+CIPMODE=1
```

响应：

OK



6. 发送数据：

```
AT+CIPSEND
```

响应：

```
> // from now on, data received from UART will be transparent transmitted to server
```

7. 退出发送数据：

在透传发送数据过程中，若识别到单独的一包数据 +++，则退出透传发送。请至少间隔 1 秒，再发下一条 AT 指令。

如果直接用键盘打字输入 +++，有可能时间太慢，不被认为是连续的三个 +，建议使用如下工具：



- 字符串输入框：+++
- 发送新行：不勾选
- 点击“发送”。

⚠ 注意：

+++ 退出透传发送数据，回到正常 AT 指令模式，TCP 连接仍然是保持的，也可以再发 AT+CIPSEND 指令，开始透传。

9. 退出透传模式：

```
AT+CIPMODE=0
```

响应：



```
OK
```

10. 删除 UDP 传输:

```
AT+CIPCLOSE
```

响应:

```
CLOSED
```

```
OK
```

7.4. 多连接 TCP 服务器

目前 AT Demo ESP32 仅支持建立一个 TCP 服务器，且必须使能多连接，即可连接多个 TCP 客户端。

以下为 ESP32 作为 SoftAP，建立 TCP 服务器的举例；如果是 ESP32 作为 Station，可在连接路由后，同理建立服务器。

1. 设置 Wi-Fi 模式:

```
AT+CWMODE=3 // SoftAP+Station mode
```

响应:

```
OK
```

2. 使能多连接:

```
AT+CIPMUX=1
```

响应:

```
OK
```

3. 建立 TCP 服务器:

```
AT+CIPSERVER=1 // default port = 333
```

响应:

```
OK
```

4. PC 连入 ESP32 SoftAP:





5. PC 作 TCP 客户端连接设备：

网络设置

(1) 协议类型
TCP Client

(2) 服务器IP地址
192.168.4.1

(2) 服务器端口
333

断开

网络数据接收

⚠ 注意：
ESP32 作为 TCP 服务器有超时机制，如果连接建立后，一段时间内无数据来往，ESP32 TCP 服务器会将 TCP 客户端踢掉。请在 PC 网络工具连上 ESP32 后建立一个 2s 的循环数据发送，用于保持连接。

6. 发送数据：

```
AT+CIPSEND=0,4           // ID number of connection is defaulted to be 0
>iopd                     // send 4 bytes to connection NO.0
                           // enter the data, no CR
```

响应：

SEND OK

⚠ 注意：
若输入的字节数目超过了指令设定的长度 (n)，则会响应 busy，并发送数据的前 n 个字节，发送完成后响应 SEND OK。

7. 接收数据：

```
+IPD,0,n:xxxxxxxxx       // received n bytes, data = xxxxxxxxxxx
```

8. 断开 TCP 连接：

```
AT+CIPCLOSE=0            // delete NO.0 connection
```

响应：

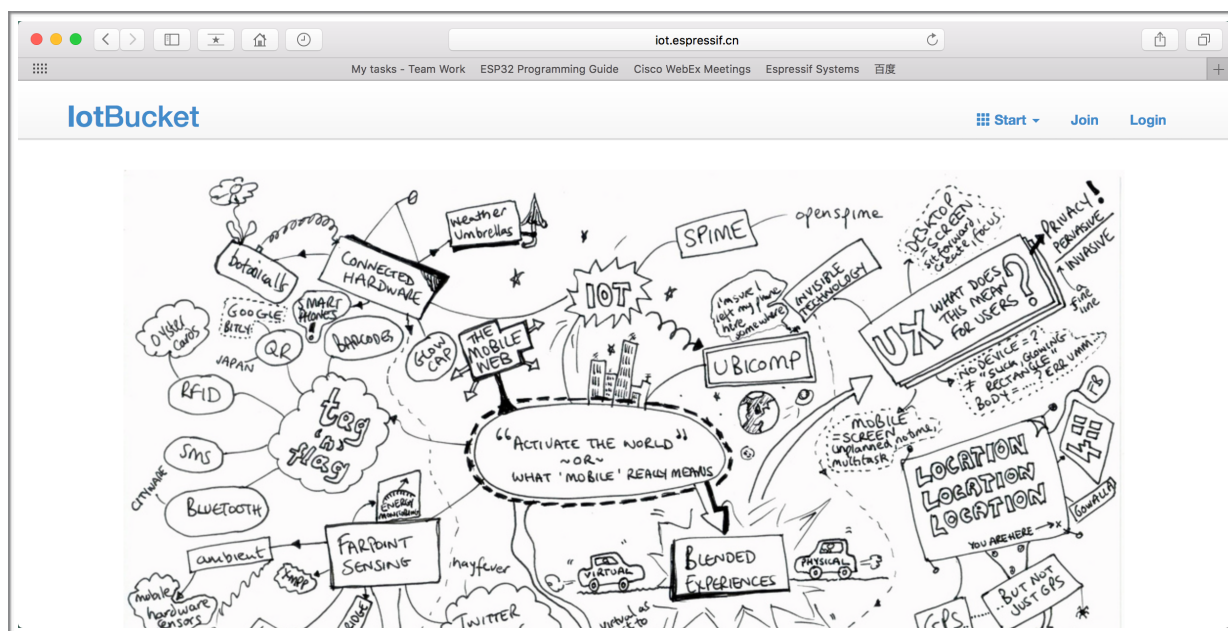
0,CLOSED
OK



8. OTA 功能

本章节主要描述如何在 iot.espressif.cn 上创建自己的设备，并自定义基于 iot.espressif.cn 的 OTA 功能。

1. 打开浏览器输入 iot.espressif.cn。



2. 点击右上角的 Join，输入用户名、邮箱和密码，注册账号。

IoT·Espressif

开始注册登录

注册

名字

Username [a-zA-Z0-9_]+

邮箱

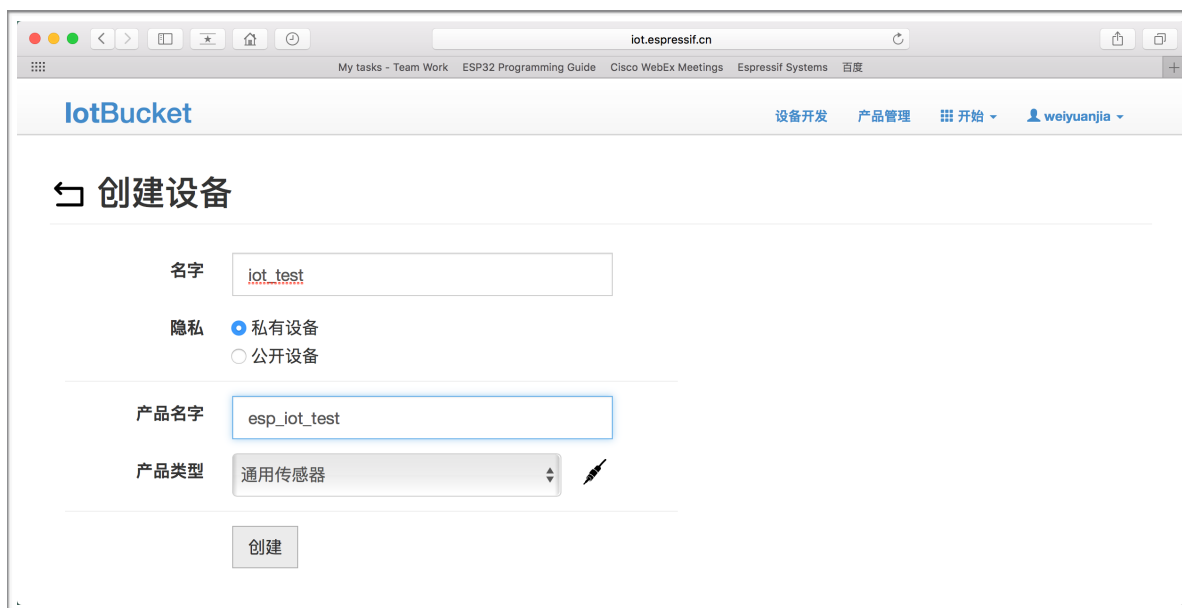
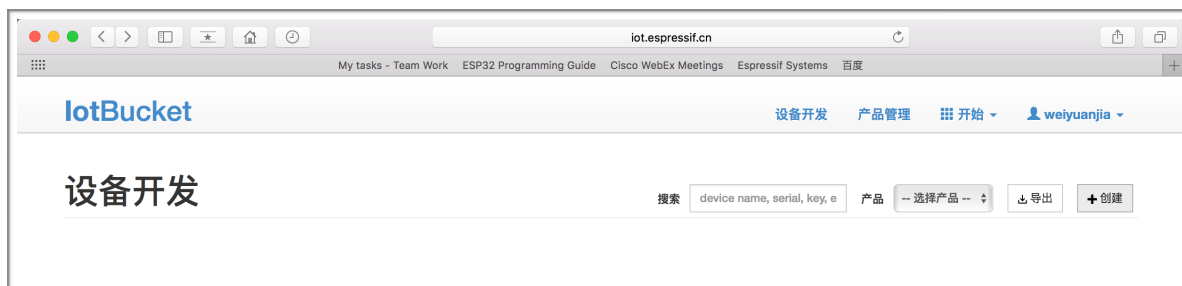
Email

密码

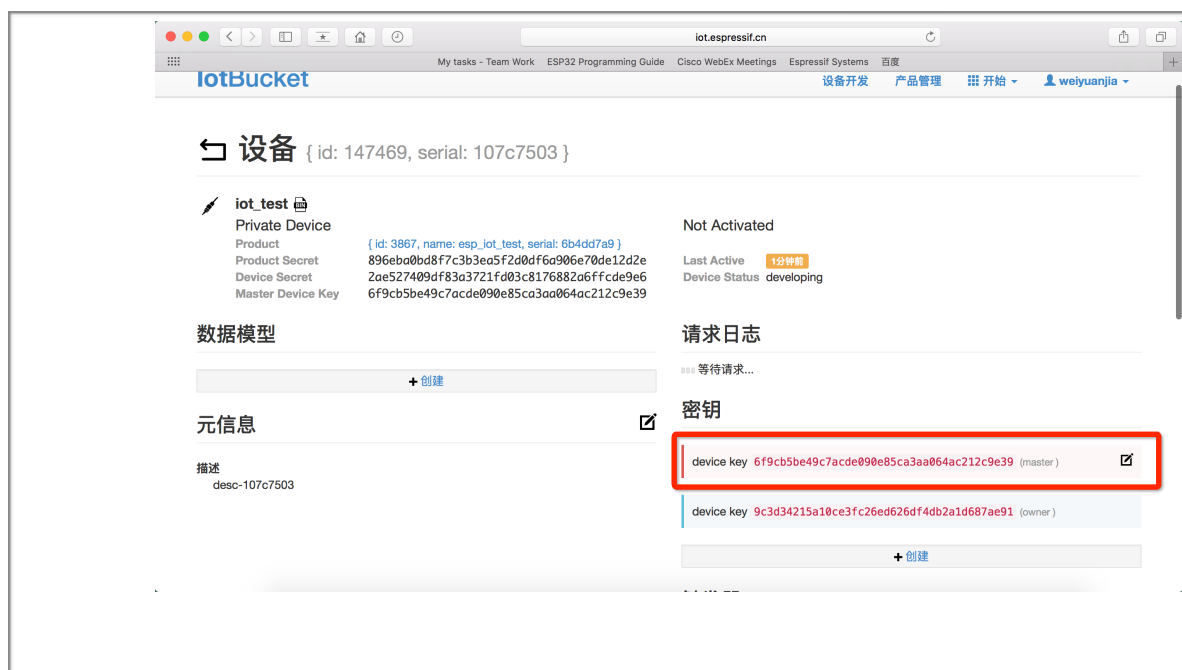
Password

注册

3. 然后点击“设备开发”里面的“创建”，创建一个自己的设备。



4. 创建成功后，会有一个密钥。



5. 使用密钥，编译自己的 OTA BIN。修改 AT OTA token 密钥的配置流程如下：



```
gcc src/crypto/dh_groups.o
xcg@xcg-linux: ~/workspace/esp_iot_at/esp-at-application-developing
/home/xcg/workspace/esp_iot_at/esp-at-application-developing/sdkconfig - Espressif IoT Development Framework

Espressif IoT Development Framework Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module
< > module capable

  SDK tool configuration --->
  Bootloader config --->
  Security features --->
  Serial flasher config --->
  Partition Table --->
  Optimization level (Debug) --->
  Component config --->

  <Select>  < Exit >  < Help >  < Save >  < Load >
```

```
gcc src/crypto/dh_groups.o
xcg@xcg-linux: ~/workspace/esp_iot_at/esp-at-application-developing
/home/xcg/workspace/esp_iot_at/esp-at-application-developing/sdkconfig - Espressif IoT Development Framework
C> Component config

Component config
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module
< > module capable

  ESP32-specific config --->
  [ ] Enable Ethernet ----
  FreeRTOS --->
  Log output --->
  LWIP --->
  mbedTLS --->
  SPI Flash driver --->
  AT --->

  英  ○  09  ⚙️

  <Select>  < Exit >  < Help >  < Save >  < Load >
```

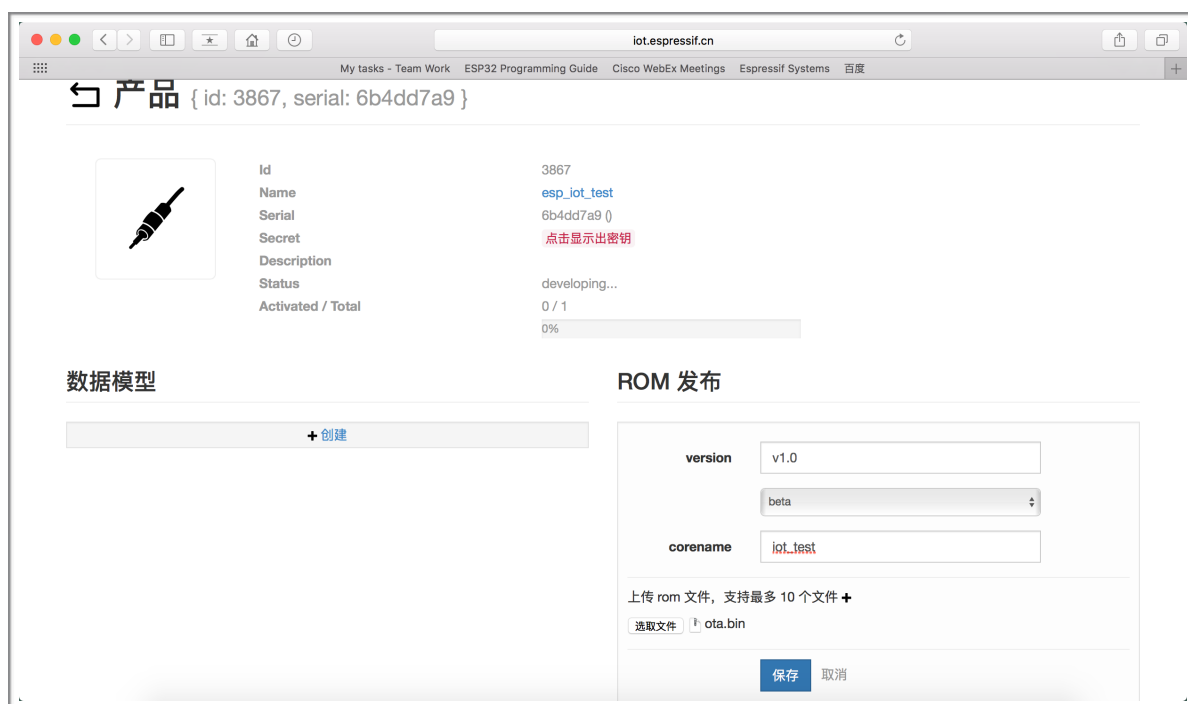
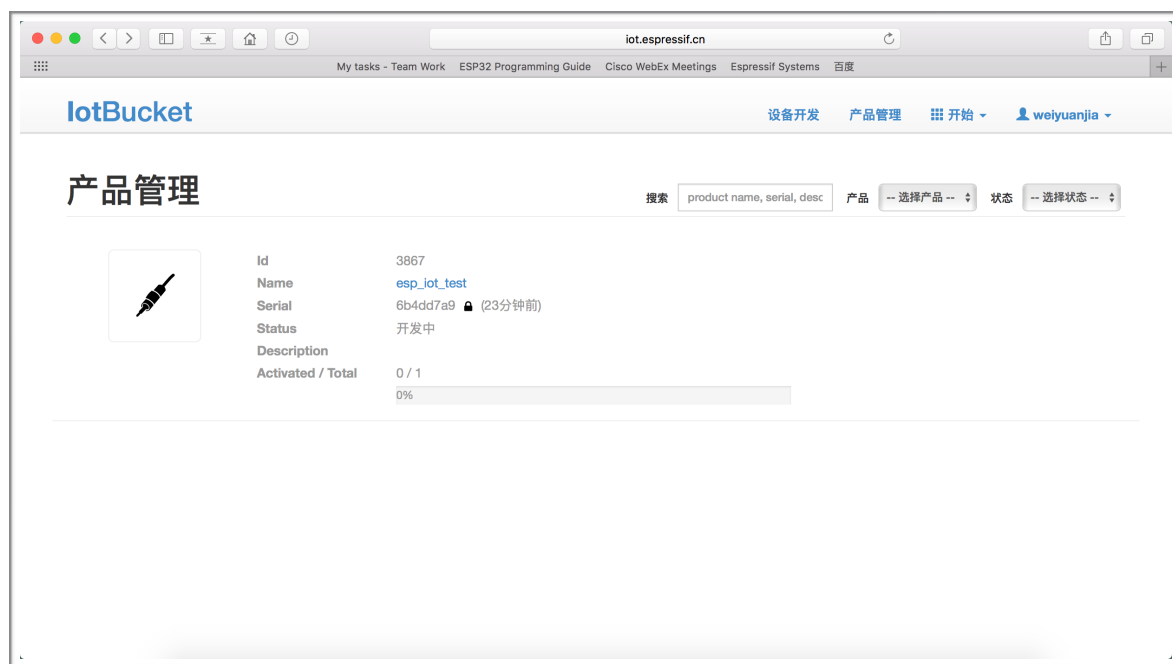
```
gcc src/crypto/dh_groups.o
xcg@xcg-linux: ~/workspace/esp_iot_at/esp-at-application-developing
/home/xcg/workspace/esp_iot_at/esp-at-application-developing/sdkconfig - Espressif IoT Development Framework
C> Component config > AT

AT
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module
< > module capable

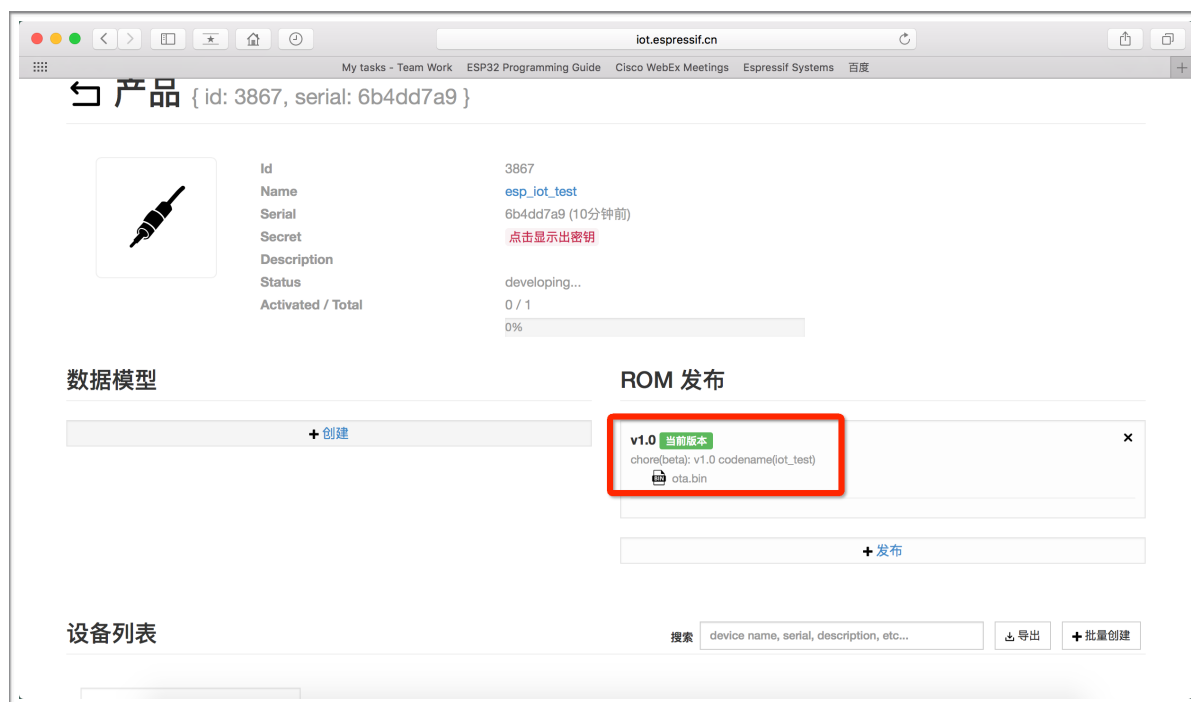
  ^(-)
  Default: UART1
  (1)  uart port number for AT command
  (16) uart rx pin for AT command
  (17) uart tx pin for AT command
  (14) uart rts pin for AT command
  (15) uart cts pin for AT command
  (iot.espressif.cn) Server IP for AT OTA.
  (80) Server port for AT OTA.
  (6f9cb5be49c7acde090e85ca3aa064ac212c9e39) The token for AT OTA.

  <Select>  < Exit >  < Help >  < Save >  < Load >
```

6. 进入产品管理界面，点击刚才创建的设备，在 ROM 发布下面，输入 version 和 corename，将上面编译的 BIN 文件重命名为 ota.bin，然后选取并点击保存。



7. 点击设置为当前版本。



8. 发送 AT+CIUPDATE, 如果网络正常, 则即可完成 OTA 升级。



9. 问题反馈

如遇到 AT 使用异常，请发邮件至[乐鑫技术支持](#)，附上如下信息：

- AT 软件的版本号，指令 AT+GMR 可获取版本信息；
- 硬件模块的信息，例如：ESP-WROOM-32；
- 测试指令步骤说明或截图，例如下图：

The screenshot shows a serial terminal window with the following settings: COM Port: 14, Baudrate: 115200, StopBits: 1, Parity: None, ByteSize: 8, Flow Control: No Ctrl Flow. The terminal output shows the following commands and responses:

```
AT+CWMODE=1
OK
AT+CIPSTA_CUR?
+CIPSTA_CUR:ip:"0.0.0.0"
+CIPSTA_CUR:gateway:"0.0.0.0"
+CIPSTA_CUR:netmask:"0.0.0.0"
OK
AT+CIPMUX=1
OK
```

- 如能提供 log 打印信息，请附上异常 log 信息，例如以下截屏：

```
Guru Meditation Error of type StoreProhibited occurred on core 0. Exception was unhandled.
Register dump:
PC      : 40135735  PS      : 00060f30  A0      : 800f913b  A1      : 3ffd66c0
A2      : 00000000  A3      : 3ffd6828  A4      : 00000b68  A5      : b33f0000
A6      : b33fffffff  A7      : 3ffb004c  A8      : 00000003  A9      : 3ffd66a0
A10     : 3ffd6828  A11     : 00000b69  A12     : 00060020  A13     : 3ffc2d30
A14     : 00000003  A15     : 00060023  SAR     : 00000000  EXCCAUSE: 0000001d
EXCVADDR: 00000038  LBEG    : 00000000  LEND    : 00000000  LCOUNT  : 00000000
Rebooting...
```




免责声明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归© 2017 乐鑫所有。保留所有权利。