

ESP8266 用户手册

Status	
Current version	V 0.3
Author	Yu Fei&Jackie
Completion Date	2014.9.2
Reviewer	
Completion Date	

CONFIDENTIAL

INTERNAL

PUBLIC

版本信息

日期	版本	撰写人	审核人	修改说明
2014.6.17	0.1	喻菲		初稿
2014.9.2	0.2	Jackie		增加章节 1.12 、1.2 和 1.3 的内容
2014.9.2	0.3	喻菲		更新 AT 指令集

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。

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© 2014 乐鑫信息技术有限公司所有。保留所有权利。

目录

版本信息.....	2
目录.....	3
1. 产品概述.....	5
1.1. 概述.....	5
1.1.1. 特点.....	6
1.1.2. 模块参数.....	7
1.1.3. 主要应用领域.....	8
1.2. 硬件介绍.....	9
1.2.1. 管脚定义.....	9
1.2.2. 电气特性.....	11
1.2.3. ESP8266 封装尺寸图.....	12
1.2.4. 开发套件.....	13
1.3. 典型应用.....	13
1.3.1. UART 应用硬件连接.....	13
1.3.2. 传感器应用硬件连接.....	13
1.3.3. 智能灯应用硬件连接.....	14
1.3.4. WIFI 智能开关应用硬件连接.....	15
2. 功能介绍.....	17
2.1. 无线组网.....	17
2.1.1. ESP8266 在 SoftAP 模式.....	17
2.1.2. ESP8266 在 station 模式.....	17
2.1.3. ESP8266 在 SoftAP + station 共存模式.....	18
2.2. 透传功能.....	18
2.3. UART 成帧机制.....	19

2.4.	加密.....	20
2.5.	省电机制.....	20
2.6.	固件升级.....	21
3.	使用指南.....	22
3.1.	Website 操作指南.....	22
3.1.1.	设备开发.....	23
3.1.2.	产品管理.....	26
3.2.	模块使用介绍.....	28
3.2.1.	软件调试工具.....	28
3.2.2.	网络连接.....	28
3.2.3.	初始参数.....	28
3.3.	应用举例.....	29
3.3.1.	无线遥控应用.....	29
3.3.2.	远程连接应用.....	29
3.3.3.	透明串口.....	30
4.	AT 指令说明.....	31
4.1.	AT 指令集概述.....	31
4.1.1.	AT 指令分类.....	31
4.1.2.	指令集.....	32
5.	其他详细资料.....	46
	附录：乐鑫联系方式.....	47

1. 产品概述

1.1. 概述

乐鑫智能互联平台——ESP8266 拥有高性能无线 SOC，给移动平台设计师带来福音，它以最低成本提供最大实用性，为 WiFi 功能嵌入其他系统提供无限可能。

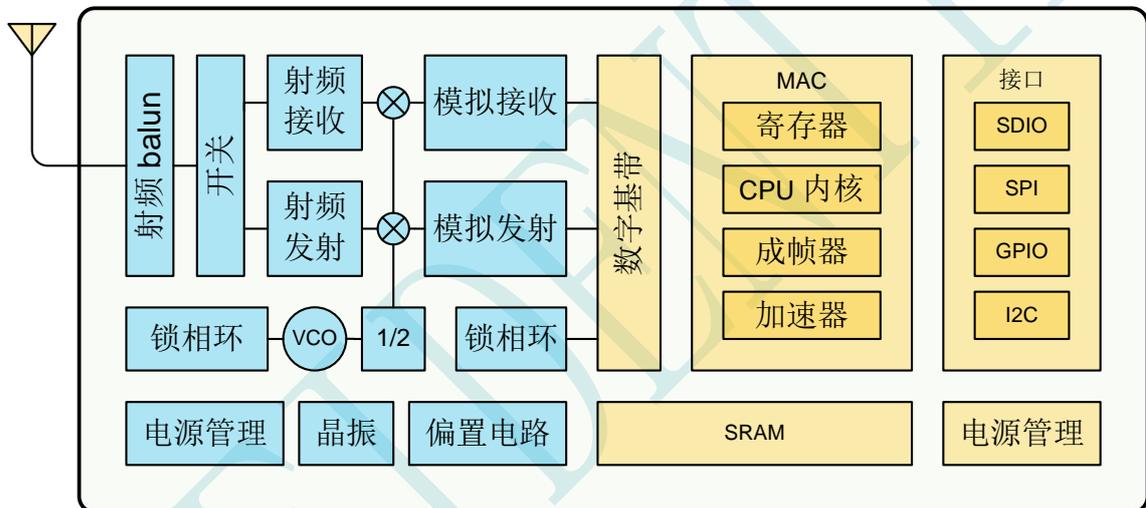


图 1: ESP8266 结构图

ESP8266 是一个完整且自成体系的 WiFi 网络解决方案，能够独立运行，也可以作为 slave 搭载于其他 Host 运行。

ESP8266 在搭载应用并作为设备中唯一的应用处理器时，能够直接从外接闪存中启动。内置的高速缓冲存储器有利于提高系统性能，并减少内存需求。

另外一种情况是，无线上网接入承担 WiFi 适配器的任务时，可以将其添加到任何基于微控制器的设计中，连接简单易行，只需通过 SPI/SDIO 接口或中央处理器 AHB 桥接口即可。

ESP8266 强大的片上处理和存储能力，使其可通过 GPIO 口集成传感器及其他应用的特定设备，实现了最低前期的开发和运行中最少地占用系统资源。

ESP8266 高度片内集成，包括天线开关balun、电源管理转换器，因此仅需极少的外部电路，且包括前端模块在内的整个解决方案在设计时将所占PCB空间降到最低。

装有ESP8266的系统表现出来的领先特征有：节能VoIP在睡眠/唤醒模式之间的快速切换、配合低功率操作的自适应无线电偏置、前端信号的处理功能、故障排除和无线电系统共存特性为消除蜂窝/蓝牙/DDR/LVDS/LCD干扰。

1.1.1. 特点

- 802.11 b/g/n
- WIFI @2.4 GHz，支持 WPA/WPA2 安全模式
- 超小尺寸模组 11.5mm*11.5mm
- 内置 10 bit 高精度 ADC
- 内置 TCP/IP 协议栈
- 内置 TR 开关、balun、LNA、功率放大器和匹配网络
- 内置 PLL、稳压器和电源管理组件
- 802.11b 模式下+ 19.5dBm 的输出功率
- 支持天线分集
- 断电泄露电流小于10uA
- 内置低功率 32 位 CPU：可以兼作应用处理器
- SDIO 2.0、 SPI、 UART
- STBC、 1x1 MIMO、 2x1 MIMO
- A-MPDU 、 A-MSDU 的聚合和 0.4 s的保护间隔
- 2ms之内唤醒、连接并传递数据包
- 待机状态消耗功率小于1.0mW (DTIM3)
- 工作温度范围 -40 ~ 125°C

1.1.2. 模块参数

分类	参数	取值	
无线参数	标准认证	CCC/FCC/CE	
	无线标准	802.11 b/g/n	
	频率范围	2.4G-2.5G(2400M-2483.5M)	
	发射功率	802.11 b:	20 dBm
		802.11 g:	17 dBm
		802.11 n:	14 dBm
	接收灵敏度	802.11 b:	(11Mbps) -91db
		802.11 g:	(54Mbps) -75db
		802.11 n:	(MCS7) -72db
天线选项	PCB板载天线、外置天线、IPEX接口天线、陶瓷贴片天线		
硬件参数	数据接口	UART	
		PWM、GPIO	
	工作电压	3.3V	
	工作电流	平均电流 80mA	
	工作温度	-40° ~125°	
	存储温度	常温	
	尺寸	5x5mm	
	外部接口	N/A	
软件参数	无线网络模式	station/softAP/SoftAP+station	
	安全机制	WPA/WPA2	
	加密类型	WEP/TKIP/AES	
	升级固件	本地串口烧录	

		云端升级
	定制开发	支持客户自定义服务器 提供 SDK 给客户二次开发
	网络协议	IPv4, TCP/UDP/HTTP/FTP
	用户配置	AT+指令集, WebSite操作, Android/iOS终端

1.1.3. 主要应用领域

- 智能电源插头
- 家庭自动化
- 网状网络
- 工业无线控制
- 婴儿监控器
- 网络摄像机
- 传感器网络
- 可穿戴电子产品
- 无线位置感知设备
- 安全ID标签
- 无线定位系统信号

1.2. 硬件介绍

1.2.1. 管脚定义

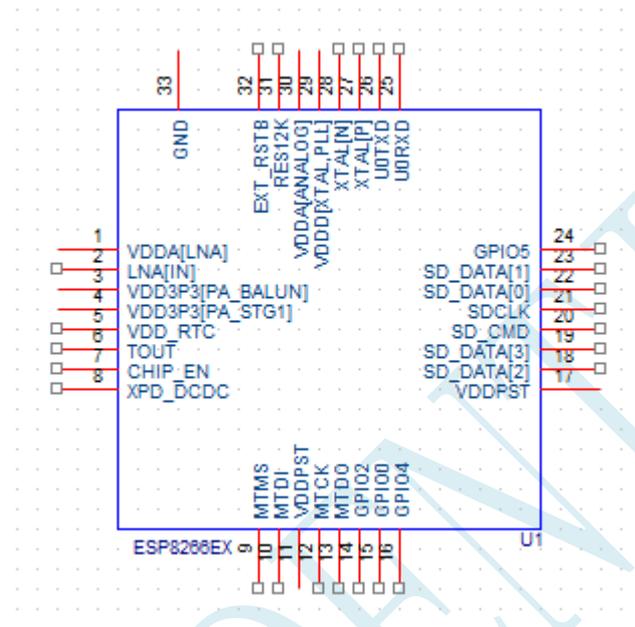


图 2: ESP8266 管脚定义图

Table 1 ESP8266 管脚功能定义

Pin	Name	Function
1	VDDA	模拟电源3.0~3.3V
2	LNA	射频天线接口，芯片输出阻抗为 50Ω，无需对芯片进行匹配，但建议保留 π 型匹配网络对天线进行匹配
3	VDD3P3	功放电源3.0~3.3V
4	VDD3P3	功放电源3.0~3.3V
5	VDD_RTC	NC(1.1V)
6	TOUT	adc 管脚
7	CHIP_EN	芯片使能端。高电平：有效，芯片正常工作；

		低电平：芯片关闭，电流很小。
8	XPD_DCDC	Deep-Sleep Wakeup; GPIO16
9	MTMS	GPIO14; HSPICLK
10	MTDI	GPIO12; HSPIQ
11	VDDPST	数字和IO电源(1.8V~3.3V)
12	MTCK	GPIO13; HSPID
13	MTDO	GPIO15; HSPICS
14	GPIO2	可用作烧写 Flash 时 UART Tx; GPIO2
15	GPIO0	GPIO0; SPICS2
16	GPIO4	GPIO4
17	VDDPST	数字和IO电源(1.8V~3.3V)
18	SDIO_DATA_2	连接到SD_D2 (串联 200Ω); SPIHD; HSPiHD
19	SDIO_DATA_3	连接到SD_D3 (串联 200Ω); SPIWP; HSPiWP
20	SDIO_CMD	连接到SD_CMD(串联 200Ω); SPICS0
21	SDIO_CLK	连接到SD_CLK (串联 200Ω); SPICLK
22	SDIO_DATA_0	连接到SD_D0 (串联 200Ω); SPIQ
23	SDIO_DATA_1	连接到SD_D1 (串联 200Ω); SPID
24	GPIO5	GPIO5
25	U0RXD	可用作烧写 Flash 时 UART Rx; GPIO3
26	U0TXD	GPIO1; SPICS1
27	XTAL_OUT	连接晶振输出端，可用于提供BT的时钟输入
28	XTAL_IN	连接晶振输入端
29	VDDD	模拟电源 3.0~3.3V
30	VDDA	模拟电源 3.0~3.3V
31	RES12K	串联 12kΩ 电阻到地
32	EXT_RSTB	外部 reset 信号(低电平有效)

注：GPIO2、GPIO0、MTDO 构成 3bit 可进行 SDIO 模式的选择。

1.2.2. 电气特性

Table 2 ESP8266 电气特性

参数	条件	最小值	典型值	最大值	单位
存放温度范围		-45		125	°C
最大焊接温度	IPC/JEDEC J-STD-020			260	°C
工作电压		0		3.8	V
任意I/O脚电压		0		3.3	V
静电释放量（人体模型）	TAMB=25°C			2	KV
静电释放量(充电设备模型)	TAMB=25°C			1	KV

Table 3 ESP8266 功耗

参数	最小值	典型值	最大值	单位
传送 802.11b, CCK 11Mbps, P OUT=+17dBm		170		mA
传送 802.11g, OFDM 54Mbps, P OUT =+15dBm		140		mA
传送 802.11n, MCS7, P OUT =+13dBm		120		mA
接收 802.11b, 包长 1024 字节, -80dBm		50		mA
接收 802.11g, 包长 1024 字节, -70dBm		56		mA
接收 802.11n, 包长 1024 字节, -65dBm		56		mA
Modem-Sleep ^①		15		mA
Light-Sleep ^②		0.9		mA
Deep-Sleep ^③		10		uA
关机		5		uA

注①: Modem-Sleep 用于需要 CPU 一直 处于工作状态 如 PWM 或 I2S 应用等。在保持 WiFi 连接时，如果没有数据传输，可根据 802.11 标准（如 U-APSD），关闭 WiFi Modem 电路来省电。例如，在 DTIM3 时，每 sleep 300mS，醒来 3mS 接收 AP 的 Beacon 包等，则整体平均电流约 15mA。

注②: Light-Sleep 用于 CPU 可暂停的应用，如 WiFi 开关。在保持 WiFi 连接时，如果没有数据传输，可根据 802.11 标准（如 U-APSD），关闭 WiFi Modem 电路并 暂停 CPU 来省电。例如，在 DTIM3 时，每 sleep 300mS，醒来 3mS 接收 AP 的 Beacon 包等，则整体平均电流约 0.9mA。

注③: Deep-Sleep 不需一直保持 WiFi 连接，很长时间才发送一次 数据包的应用，如每 100 秒测量一次温度的传感器。例如，每 300S 醒来后需 0.3~1s 连上 AP 发送数据,则整体平均电流可远小于 1mA。

1.2.3. ESP8266 封装尺寸图

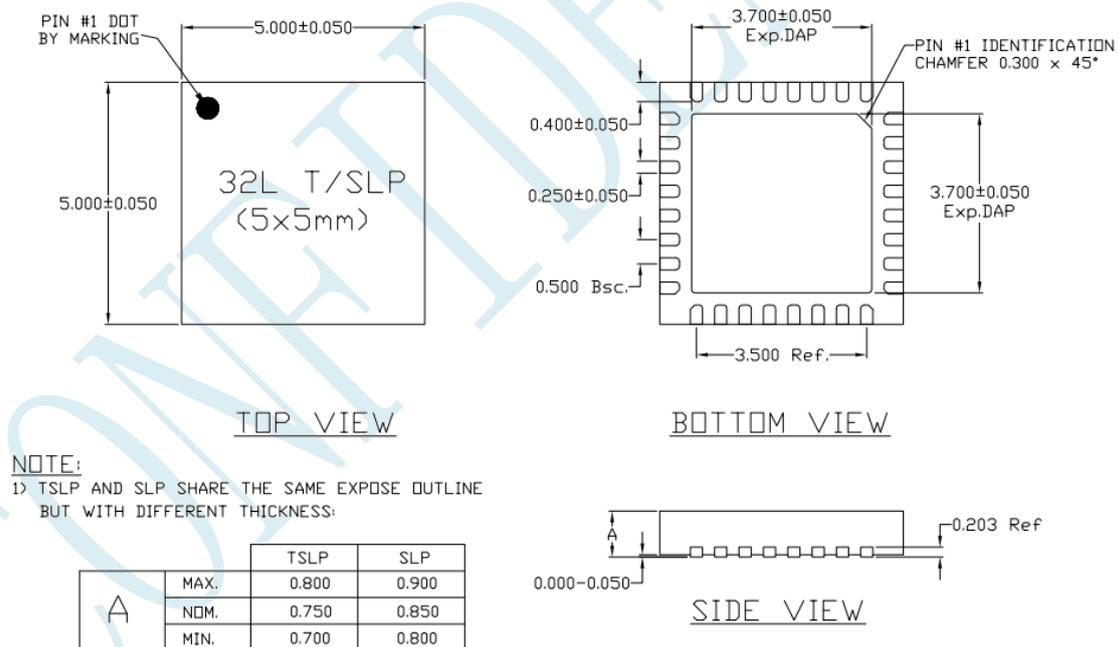


图 3: ESP8266 封装尺寸图

1.2.4. 开发套件

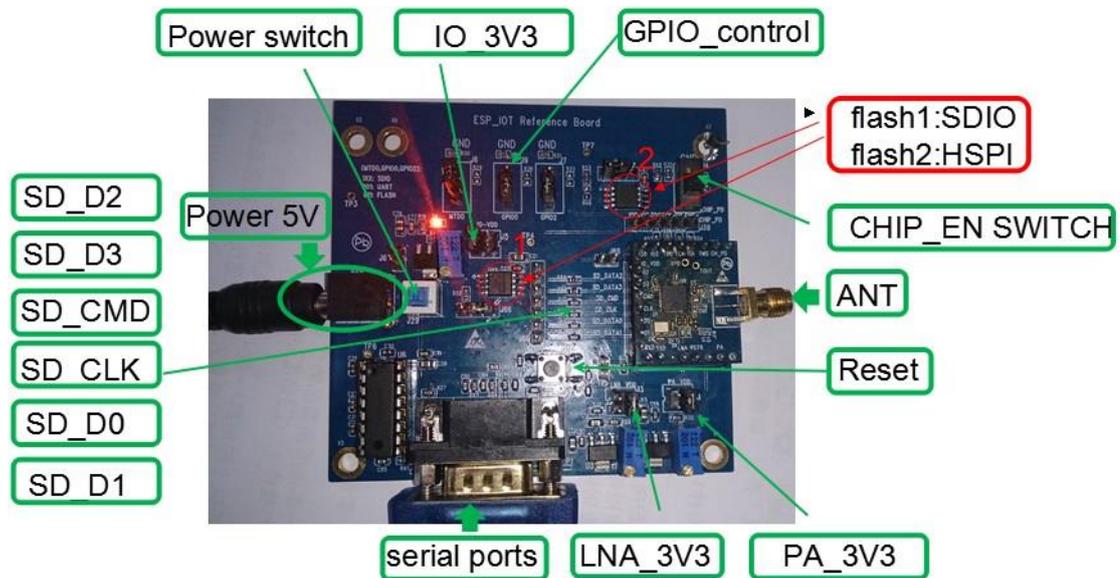


图 4: ESP8266 开发板示意图

1.3. 典型应用

1.3.1. UART 应用硬件连接

我司已根据UART应用定义以下pin脚:

UART0: (PIN 25) U0RXD+ (PIN 26) U0TXD---通信

UART1: (PIN 14) GPIO2(TXD)---打印

目前 uart0 可以用来收发用户自己的数据包, uart1 用作打印信息。

已应用方案:

如图4所示ESP8266开发板

1.3.2. 传感器应用硬件连接

我司已根据传感器应用定义以下pin脚:

(PIN 9) MTMS---I2C_SCL;

(PIN 14) GPIO2---I2C_SDA;

(PIN 12) MTCK---复位按键 (按住复位按键重新上电，可完成复位);

(PIN 15) GPIO0---wifi 工作状态指示灯;

(PIN 10) MTDI--- 与服务器通信的指示灯;

(PIN 25) U0RXD---Button，暂未定义功能;

(PIN 13) MTDO---LED，暂未定义功能。

已应用方案:

如图5所示我司传感器应用demo板



图 5: 基于 ESP8266 的传感器应用硬件示意图

1.3.3. 智能灯应用硬件连接

我司已根据智能灯应用定义以下pin脚:

(PIN 9) MTMS---红外接收;

三路 PWM 输出:

(PIN 10) MTDI---红色灯控制;

(PIN 13) MTDO---绿色灯控制;

(PIN 12) MTCK---蓝色灯控制;

已应用方案:

如图6所示我司智能灯应用demo板



图 6：基于 ESP8266 的智能灯应用硬件示意图

1.3.4. WIFI 智能开关应用硬件连接

我司已根据WIFI智能开关应用定义以下pin脚：

(PIN 13) MTDO---控制继电器，可高低电平控制；

(PIN 15) GPIO0--- 指示 wifi 工作状态；

(PIN 12) MTCK---接复位按键(长按 5s 可完成复位功能)；

已应用方案：

如图7所示我司WIFI智能开关应用demo板

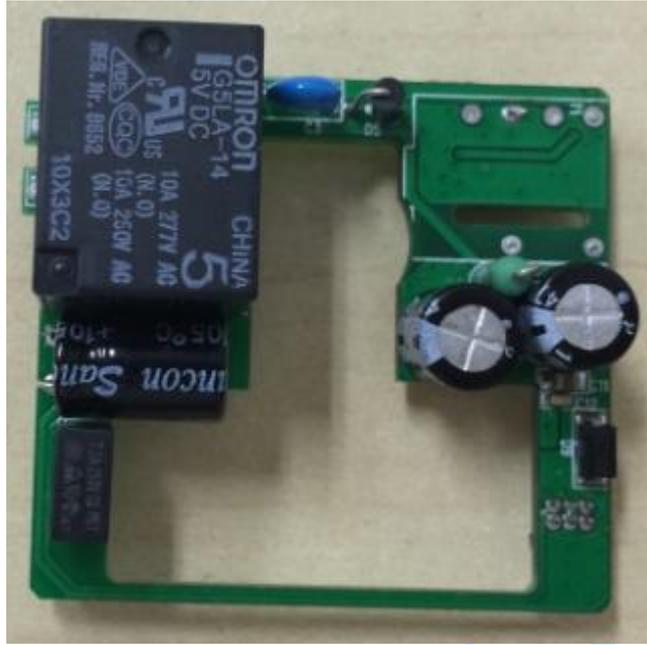


图 7：基于 ESP8266 的 WIFI 智能开关应用硬件示意图
(我司芯片在背面)

2. 功能介绍

2.1. 无线组网

ESP8266 支持 softAP 模式, station 模式, softAP + station 共存模式三种。
利用 ESP8266 可以实现十分灵活的组网方式和网络拓扑。

注,

SoftAP: 即无线接入点, 是一个无线网络的中心节点。通常使用的无线路由器就是一个无线接入点。

Station: 即无线终端, 是一个无线网络的终端端。

2.1.1. ESP8266 在 SoftAP 模式

ESP8266 作为 softAP, 手机、电脑、用户设备、其他 ESP8266 station 接口等均可以作为 station 连入ESP8266, 组建成一个局域网。



2.1.2. ESP8266 在 station 模式

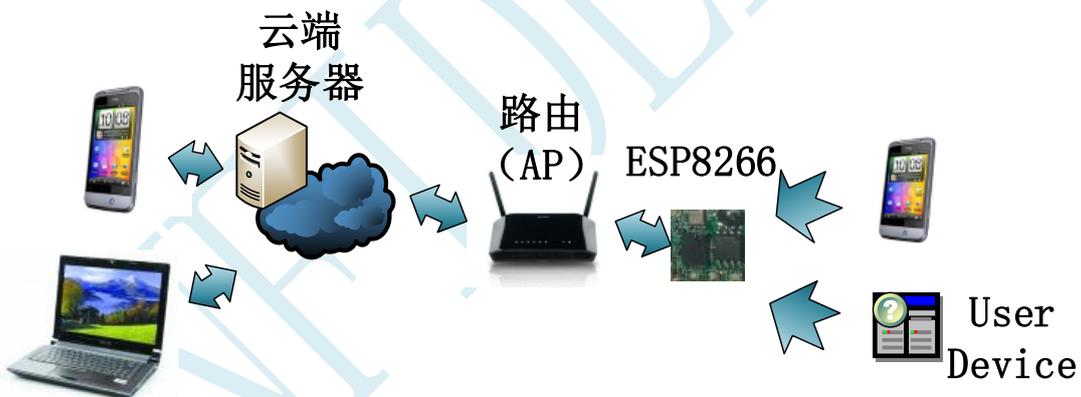
ESP8266 作为 station, 通过路由器 (AP) 连入 internet, 可向云端服务器上传、下载数据。用户可随时使用移动终端 (手机、笔记本等), 通过云端

监控 ESP8266 模块的状况，向 ESP8266 模块发送控制指令。



2.1.3. ESP8266 在 SoftAP + station 共存模式

ESP8266 支持 softAP+station 共存的模式，用户设备、手机等可以作为 station 连入 ESP8266 的 softAP 接口，同时，可以控制 ESP8266 的 station 接口通过路由器 (AP) 连入 internet。



2.2. 透传功能

透传，即透明传输功能。Host 通过 uart 将数据发给 ESP8266，ESP8266 再通过无线网络将数据传出去；ESP8266 通过无线网络接收到的数据，同理通过 uart 传到 Host。ESP8266 只负责将数据传到目标地址，不对数据进行处理，发送方和接收方的数据内容、长度完全一致，传输过程就好像透明一样。

透传需要先建立连接:

- 无线网络参数:
 - ◇ 网络名称 (SSID)
 - ◇ 安全模式
 - ◇ 密钥 (password)
- TCP 连接参数
 - ◇ 协议类型
 - ◇ 连接类型 (client)
 - ◇ 目的 IP 地址
 - ◇ 目的端口
- 串口参数
 - ◇ 波特率
 - ◇ 数据位
 - ◇ 检验位
 - ◇ 停止位
 - ◇ 硬件流控

2.3. UART 成帧机制

ESP8266 判断 UART 传来的数据时间间隔, 若时间间隔大于 20ms, 则认为一帧结束; 否则, 一直接收数据到上限值 2KB, 认为一帧结束。ESP8266 模块判断 UART 来的数据一帧结束后, 通过 WIFI 接口将数据转发出去。

成帧时间间隔为 20ms, 一帧上限值为 2KB。

2.4. 加密

ESP8266 支持多种无线网络加密方式，包括：

- WEP (only station)
- WPA-PSK/TKIP
- WPA-PSK/AES
- WPA2-PSK/TKIP
- WPA2-PSK/AES

2.5. 省电机制

ESP8266 IOT 支持三种省电模式：

模式	Modem-Sleep	Light-Sleep	Deep-Sleep
动作	关闭 WiFi Modem 电路；CPU 和其他外设正常运行。	关闭 WiFi Modem 电路、晶振和 PLL；CPU 和其他外设处于时钟暂停待机状态。	仅 RTC 电路工作，关闭其他电路，芯片处于极低功耗待机状态。
电流	10~20mA	0.5mA	10~20uA
唤醒	可唤醒	可唤醒	无法唤醒，设备依照设定，定时醒来。
应用场景	用于 CPU 需要一直工作的场景。 如 PWM 或 I2S 应用等。 如果没有数据传输，可根据 802.11 标准	用于 CPU 可暂停的应用。 如 WiFi 开关。 如果没有数据传输，可根据 802.11 标准（如 U-APSD），关	用于不需一直保持 WiFi 连接，很长时间才发送一次数据包的应用。 如每 100 秒测量一次温度的传感器。

	<p>(如 U-APSD)，关闭 WiFi Modem 电路来省电。</p> <p>例如，在 DTIM3 时，每 sleep 300ms，醒来 3ms 接收 AP 的 Beacon 包等，整体平均电流约 15mA。</p>	<p>关闭 WiFi Modem 电路，并暂停 CPU 来省电。</p> <p>例如，在 DTIM3 时，每 sleep 300ms，醒来 3ms 接收 AP 的 Beacon 包等，则整体平均电流约 0.9mA。</p>	<p>例如，每 300S 醒来后需 0.3~1s 连上 AP 发送数据，则整体平均电流可远小于 1mA。</p>
--	---	---	--

2.6. 固件升级

ESP8266 除了传统的串口烧录方式，还支持云端升级的方式来更新固件。只需将新版固件上传至服务器，在 ESP8266 联网的情况下，服务器会推送更新消息到用户，用户可自行选择是否升级。

串口烧录升级，请参见文档“Espressif IoT SDK 使用手册”。

云端升级，请参见文档“云端升级实现方案”。

3. 使用指南

3.1. Website 操作指南

1) 打开网址 <http://iot.espressif.cn/#/>

“开始”->“开发者API”： 设备控制的相关API说明。

“开始”->“帮助”： 使用帮助示例说明，用户可参照创建自己的产品。

“注册”： 新用户注册。

“登录”： 已注册用户登录。



2) 点击“设备开发”显示用户常用于调试的设备列表，针对开发者。

“搜索”： 输入设备名称或device key等信息查找设备。

“导出”： 用于导出设备列表。

“创建”： 创建一个新设备。

设备开发

 搜索 产品 --选择产品-- 导出 创建

 test-demo Id 127 Serial 5c396421 4天前	 device-name-608af4eb Id 209 Serial 608af4eb 4天前	 tm Id 210 Serial acbd414b 4天前	 device-name-a69ee7c4 Id 211 Serial a69ee7c4 4天前
 aaa Id 212	 device-name-6677895f Id 213	 light001 Id 55	 device-name-637b9f0b Id 243

- 3) “产品管理”显示用户的产品列表，适用于商业管理各类产品。可通过“搜索”、“产品”、“状态”设置过滤条件，查询产品。

产品管理

 搜索 产品 --选择产品-- 状态 --选择状态--

	Id 1 Name PowerPlugProduct Serial f3ec8e37 (1小时前) Status deployed Description description Of product(serial: f3ec8e37) Activated / Total 69 / 191 6.12565445026178%
	Id 3 Name light Serial 9f4f24e2 (3小时前)

3.1.1. 设备开发

针对设备，可以查看设备历史数据，增添定时器，云端升级，自定义反向控制。

3.1.1.1. 历史数据

在设备页面 -> “数据模型” -> “数据”，可查看设备的历史数据。

数据模型		维度
plug-status		一维
单位	符号	标签
boolean	on/off	plug
描述 可选		
plug-status1:on,0:off		
显示 · 图像 · 数据 · 编辑 · 删除		
#	at	x
34645	2014-07-01 09:43:03	0
34643	2014-07-01 09:42:24	1

3.1.1.2. 调试信息

“请求日志”记录了服务器收到的指令，用于设备调试。

请求日志

POST	/v1/ping/	2014-06-10 12:24:35	▼
POST	/v1/datastreams/plug-status/datapoint/	2014-06-10 12:22:49	▼
POST	/v1/datastreams/plug-status/datapoint/	2014-06-10 12:22:43	▼
POST	/v1/ping/	2014-06-10 12:22:39	▼
POST	/v1/ping/	2014-06-10 12:21:44	▼
POST	/v1/ping/	2014-06-10 12:20:52	▼

3.1.1.3. 定时器

Espressif 支持三种定时器：

- 1) 精确时间：在特定的某时间，执行某动作。
- 2) 周期循环：每隔特定时长，执行一遍某动作，循环往复。
- 3) 按周循环：每周的某几天的某固定时间，执行某动作，循环往复。

Timer

在 20140709131520 执行动作 action1

每 4 hour 执行动作 action2

每周 [1,3,5]

在 161000 执行动作 action3

类型：

精确时间 周期循环 按周循环

时间

动作

+

20141011121314

保存

保存到设备

取消

3.1.1.4. 自定义的反向控制

用户可以自定义动作，反向控制设备。

IoT Bucket

产品管理



Id	1
Name	humiture
Serial	f3ec8e37 (3天前)
Status	deployed
Description	description Of product(serial: f3ec8e37)
Activated / Total	86 / 552
	15.579710144927537%



Id	3
Name	light
Serial	9f4f24e2 (3天前)
Status	developing...

- 2) 为某一产品更新固件，只需上传至服务器，Espressif server 会向各个设备推送固件升级消息，用户可自行选择是否升级。

ROM 发布

v1.1 当前版本

chore(beta): v1.1

user1.bin

user2.bin

v1.0 设置为当前版本

chore(beta): v1.0 codename(test)

user1.bin

user2.bin

3.2. 模块使用介绍

3.2.1. 软件调试工具

Espressif 推荐使用如下一些工具作为ESP8266的调试工具。（用户也可以自行选择其他同类型的工具。）

- 烧录工具： FLASH_DOWNLOAD_TOOLS.exe
- 串口传输工具： SecureCRTPortable.exe
- 网络调试工具： NetAssist.exe

3.2.2. 网络连接

ESP8266 有两种组网接口，softAP 接口和 station 接口，且两种接口可同时并存使用。

用户按照实际需求应用：

- softAP 接口

Phone 或 PC 作为 station，连入 ESP8266 的 softAP 接口，如需调试，可用 PC 连接 ESP8266 的串口查看 log 信息。

- station 接口

ESP8266 作为 station，连入无线路由（AP），如需调试，可用 PC 连接 ESP8266 的串口查看 log 信息。

3.2.3. 初始参数

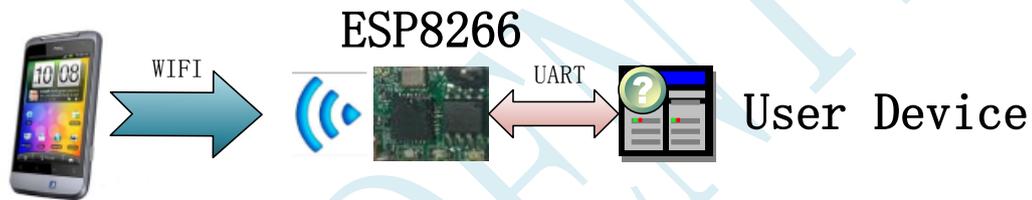
- 模块默认SSID 为： ESP_XXXXXX (XXXXXX 为模块 MAC 地址后6位)
- 默认加密方式： WPA/WPA2

- 用户串口参数默认为：74880，8，1，None
- 模块作为 softAP，默认IP：192.168.4.1

3.3. 应用举例

3.3.1. 无线遥控应用

Phone 作为 station，连入 ESP8266 提供的 softAP，ESP8266 可通过 UART 与设备相连，进行无线控制。



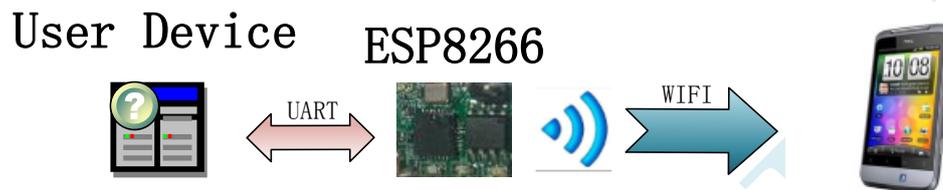
3.3.2. 远程连接应用

ESP8266 作为 station，通过无线路由（AP）连入 internet，可以向云端服务器上传数据、下载固件更新；移动终端（手机、笔记本等）也可以通过云端服务器，远程控制 ESP8266。



3.3.3. 透明串口

用户设备的主 IC 作为 Host，通过 UART 向 ESP8266 发送数据，ESP8266 再将数据通过无线网络传输给目的地址，实现透明传输的功能。



4. AT 指令说明

描述 Espressif ESP8266 模块 AT 指令集功能以及使用方法。

4.1. AT 指令集概述

指令集分为：基础 AT 命令、Wifi 功能 AT 命令、TCP/IP 工具箱 AT 命令等。

4.1.1. AT 指令分类

分类	指令格式	指令功能
测试命令	AT+<x>=?	该命令用于查询设置命令或内部程序设置的参数以及其取值范围。
查询命令	AT+<x>?	该命令用于返回参数的当前值。
设置命令	AT+<x>=<...>	该命令用于设置用户自定义的参数值。
执行命令	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：

- 1) 不是每条 AT 指令都具备上述 4 类命令。
- 2) 后文指令中[]内数据为缺省值，不必填写或可能不显示。
- 3) 波特率 115200。
- 4) 参数为字符串的，需要加上双引号。例如：

```
AT+CWSAP="ESP_756190","20130826",1,4
```

4.1.2. 指令集

命令	描述
基础指令	
AT	测试 AT 启动
AT+RST	重启模块
AT+GMR	查看版本信息
wifi 功能指令	
AT+CWMODE	选择 WIFI 应用模式
AT+CWJAP	加入 AP
AT+CWLAP	列出当前可用 AP
AT+CWQAP	退出与 AP 的连接
AT+CWSAP	设置 AP 模式下的参数
AT+ CWLIF	查看已接入设备的 IP
TCP/IP 指令	
AT+CIPSTATUS	获得连接状态
AT+CIPSTART	建立 TCP 连接或注册 UDP 端口号
AT+CIPSEND	发送数据
AT+CIPCLOSE	关闭 TCP 或 UDP
AT+CIFSR	获取本地 IP 地址
AT+CIPMUX	启动多连接
AT+CIPSERVER	配置为服务器
AT+CIPMODE	设置模块传输模式
AT+CIPSTO	设置服务器超时时间
接收的数据	
+IPD	接收到的数据

4.1.2.1. AT

AT: 测试 AT 启动		
执行指令:	响应	OK
AT	说明	None

4.1.2.2. AT+RST

AT+RST: 重启模块		
执行指令:	响应	OK
AT+RST	说明	None

4.1.2.3. AT+GMR

AT+GMR: 查看版本信息		
执行指令:	响应	<number>
AT+GMR		OK
	说明	< number >: 8 位版本号

4.1.2.4. AT+CWMODE

AT+CWMODE: 选择 WIFI 应用模式		
测试指令:	响应	+CWMODE:(<mode>取值列表)
AT+CWMODE=?		OK
	说明	响应返回当前可支持哪些模式?

查询命令： AT+CWMODE?	响应	+CWMODE:<mode> OK
	说明	响应当前处于哪种模式？
设置指令： AT+CWMODE=<mode>	响应	OK
	说明	此指令需重启后生效(AT+RST)。 指令参数<mode>： 1 - Station 模式； 2 - AP 模式； 3 - AP + Station 共存模式。

4.1.2.5. AT+CWJAP

AT+CWJAP: 加入 AP		
查询命令： AT+ CWJAP?	响应	+ CWJAP:<ssid> OK
	说明	响应返回当前选择的 AP
设置指令： AT+ CWJAP =<ssid>,<pwd>	响应	OK 或 ERROR
	说明	指令参数： <ssid>字符串参数，接入点名称 <pwd>字符串参数，密码最长 64 字节 ASCII

4.1.2.6. AT+CWLAP

AT+CWLAP: 列出当前可用 AP		
执行指令: AT+CWLAP	响应	成功, 返回AP列表 + CWLAP: <ecn>,<ssid>,<rssi> OK 或者 失败, 返回 ERROR
	说明	响应参数说明: <ecn>0 OPEN 1 WEP 2 WPA_PSK 3 WPA2_PSK 4 WPA_WPA2_PSK <ssid>字符串参数, 接入点名称 <rssi>信号强度

4.1.2.7. AT+CWQAP

AT+CWQAP: 退出与 AP 的连接		
测试指令:	响应	OK
AT+CWQAP=?	说明	None
执行指令:	响应	OK

AT+ CWQAP	说明	None
-----------	----	------

4.1.2.8. AT+ CWSAP

AT+CWSAP: 设置 AP 模式下的参数		
测试指令	响应	
	说明	
查询命令: AT+CWSAP?	响应	返回当前 AP 参数 +CWSAP:<ssid>,<pwd>,<chl>,<ecn>
	说明	None
设置指令: AT+CWSAP= <ssid>,<pwd>,<chl>, <ecn>	响应	OK 或 ERROR
	说明	指令参数: <ssid>字符串参数, 接入点名称 <pwd>字符串参数, 密码最长 64 字节 ASCII <chl>通道号 <ecn>加密方式 0 OPEN 1 WEP 2 WPA_PSK 3 WPA2_PSK 4 WPA_WPA2_PSK
说明	该组指令只有在 AP 模式开启后有效	

4.1.2.9. AT+CWLIF

AT+CWLIF: 查看已接入设备的 IP		
执行指令: AT+ CWLIF	响应	<ip addr> OK
	说明	<ip addr> : 已接入设备的 IP 地址

4.1.2.10. AT+ CIPSTATUS

AT+CIPSTATUS: 获得连接状态		
测试指令	响应	OK
	说明	None
执行指令: AT+ CIPSTATUS	响应	返回当前模块的连接状态和连接参数。 STATUS:<stat> + CIPSTATUS:<id>,<type>,<addr>,<port>,<tetype> OK
	说明	响应值说明: <id>连接的 id 号 0-4 <type>字符串参数, 类型 TCP 或 UDP <addr>字符串参数, IP 地址 <port>端口号 <tetype> 0: 本模块做 client 的连接 1: 本模块做 server 的连接

4.1.2.11. AT+CIPSTART

AT+CIPSTART: 建立 TCP 连接或注册 UDP 端口号		
测试指令 AT+CIPSTART=?	响应	1) 若设置 AT+CIPMUX=0 +CIPSTART:(<type>取值列表),(<IP address>范围),(<port>范围) +CIPSTART:(<type>取值列表),(<domain name>范围),(<port>范围) OK 2) 若设置 AT+CIPMUX=1 +CIPSTART:(id),(<type>取值列表),(<IP address>范围),(<port>范围) +CIPSTART: (id), (<type>取值列表),(<domain name>范围),(<port>范围)
	说明	None
设置命令 1)单路连接 (+CIPMUX=0) AT+CIPSTART=	响应	OK - 格式正确且连接成功 或者 ERROR – 失败 或者 ALREAY CONNECT - 连接已存在
<type>,<addr>,<port> 2)多路连接		指令参数: <id> 0-4 连接的 id 号 <type>字符串参数, 表明连接类

(+CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	说明	型。“TCP”建立 tcp 连接；“udp”建立 UDP 连接 <addr>字符串参数，远程服务器 IP 地址 <port>远程服务器端口号
说明		

4.1.2.12. AT+CIPSEND

AT+CIPSEND: 发送数据		
测试指令： AT+CIPSEND=?	响应	OK
	说明	None
设置指令： 1)单路连接时 (+CIPMUX=0) AT+CIPSEND=<length> 2)多路连接时 (+CIPMUX=1) AT+CIPSEND= <id>,<length>	响应	响应返回指定长度的数据。 收到指令后先换行返回">", 然后开始接收串口数据，当数据长度满 length 时发送数据。 如果未建立连接或连接被断开，返回 ERROR 如果数据发送成功，返回 SEND OK
	说明	指令参数： <id>需要用于传输连接的 id 号 <length>数字参数，表明发送数据的

		长度，最大为 2048
执行指令： AT+CIPSEND 只在透传模式下有效	响应	收到指令后先换行返回">" 然后就进入了透传模式，每包数据以 20ms 间隔区分，每包最大 2048 字节。 当输入单独一包"+++"返回指令模式。
	说明	该指令必须在开启透传且为单连接的情况下使用。

4.1.2.13. AT+CIPCLOSE

AT+CIPCLOSE: 关闭 TCP 或 UDP		
测试指令：	响应	OK
AT+CIPCLOSE=?	说明	None
设置命令： 多路连接时 AT+CIPCLOSE=<id>	响应	如果输入正确，返回 OK 如果没有该连接则，返回 Link is not
	说明	指令参数： <id>需要关闭的连接 id 当 id 为 5 时关闭所有连接（开启 server 后 id 为 5 无效）

执行指令： 单路连接时 AT+CIPCLOSE	响应	如果输入正确，返回 OK 如果没有连接则，返回 ERROR
	说明	None
说明	关闭后的状态为 unlink	

4.1.2.14. AT+CIFSR

AT+CIFSR: 获取本地 IP 地址		
测试指令： AT+CIFSR=?	响应	OK
	说明	None
执行命令： AT+ CIFS	响应	+CIFSR:<IP address> OK ERROR
	说明	AP 模式下该指令无效。 响应返回： <IP address> - 本机当前 IP 地址(station)

4.1.2.15. AT+ CIPMUX

AT+CIPMUX: 启动多连接		
	响应	+CIPMUX:<mode>

查询命令： AT+CIPMUX?		OK
	说明	None
设置指令： AT+CIPMUX=<mode>	响应	OK 如果已经处于连接状态则，返回 Link is builded
	说明	只有当连接都断开后才能更改，如果开启过 server 需要重启模块。 指令参数： <mode> 0 单路连接模式 1 多路连接模式

4.1.2.16. AT+ CIPSERVER

AT+CIPSERVER: 配置为服务器		
设置指令: AT+CIPSERVER= <mode>[,<port>]	响应	开启 server 后自动建立 server 监听 OK 关闭 server 需要重启
	说明	指令参数: <mode> 0 关闭 server 模式 1 开启 server 模式 <port>端口号, 缺省值为 333
说明	开启 server 后自动建立 server 监听 当有 client 接入会自动按顺序占用一个连接 AT+ CIPMUX=1 时才能开启服务器	

4.1.2.17. AT+ CIPMODE

AT+CIPMODE: 选择传输模式		
查询命令: AT+ CIPMODE?	响应	+CIPMODE:<mode> OK
	说明	响应当前处于哪种模式?
设置指令: AT+ CIPMODE =<mode>	响应	OK 如果已经处于连接状态, 则返回 Link is builded
	说明	指令参数<mode>: 0 – 非透传模式; 1 – 透传模式;

4.1.2.18. AT+ CIPSTO

AT+CIPSTO: 设置服务器超时时间		
查询命令: AT+ CIPSTO?	响应	+CIPSTO:<time> OK
	说明	响应当前超时时长
设置指令: AT+ CIPMODE =<mode>	响应	OK
	说明	< time>: 0~28800 服务器超时时间, 单位为 s

4.1.2.19. +IPD 接收到的网络数据

+IPD: 接收到网络数据	
<p>参考</p> <p>1)单路连接时 (+CIPMUX=0) +IPD,<len>:<data></p>	<p>说明</p> <p>此指令是模块发出指令，当模块接收到网络数据时向串口发送+IPD 和数据</p>
<p>1)多路连接时 (+CIPMUX=1) +IPD,<id>,<len>:<data></p>	<p><id>收到连接的 id 号</p> <p><len>数据长度</p> <p><data>收到的数据</p>

5. 其他详细资料

如需基于 ESP8266 模块做二次开发，我司将提供更多详细的参考资料，以下为部分参考文档列表：

- **Espressif IoT SDK 使用手册**
介绍 ESP8266 的开发环境如何搭建。
- **Espressif IoT Demo 使用手册**
介绍 ESP8266 IOT Demo 如何调试使用。
- **Espressif IoT SDK 编程手册**
介绍 ESP8266 IOT SDK 提供的所有软件接口。
- **Espressif Cloud Introduction**
介绍 Espressif 的云端服务如何使用。
- **Espressif Flash Operation**
技术细节文档，介绍 ESP8266 Flash 读写操作。
- **Espressif 定时器功能**
技术细节文档，介绍 ESP8266 基于云端服务的定时功能如何实现。
- **云端升级实现方案**
技术细节文档，介绍 ESP8266 基于 WIFI 的云端升级如何实现。

附录：乐鑫联系方式

.....
地址：上海市浦东张江高科技园区碧波路 456 号 A201 室 邮编：201203

电话：（86）021-61062079

网址：<http://www.espressif.com/>

邮件联系：sales@espressif.com
.....

了解更多关于乐鑫信息科技的资讯，请访问网站：<http://www.espressif.com/>

<结束>