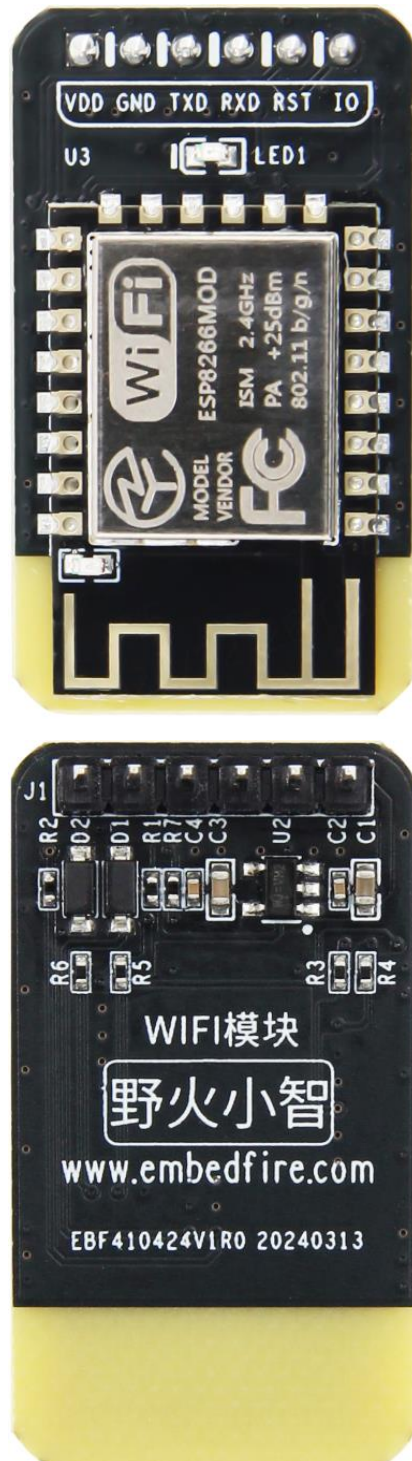


WIFI_ESP8266-12F 模块



销售与服务联系

东莞野火科技有限公司

地址：东莞市大岭山镇石大路 2 号艺华综合办公大楼 301 1 2 3 4 楼

官网：<https://embedfire.com>

论坛：<http://www.firebbs.cn>

资料：<https://doc.embedfire.com>

天猫：<https://yehuosm.tmall.com>

京东：<https://yehuo.jd.com/>

邮箱：embedfire@embedfire.com

电话：0769-33894118

扫码获得更多精彩



野火百科



野火电子



野火天猫店



野火京东店



野火抖音号



野火视频号



野火B站号



野火小师妹

第一章 产品介绍

1.1 模块简介

WIFI_ESP8266-12F 模块是基于 ESP-12F 模组设计的一款串口转 WIFI 模块，通过串口与主控制器进行通信，支持串口透传，支持串口 AT 指令集，集成 WIFI 无线网络连接功能和 TCP/IP 协议栈，用户通过串口即可实现网络访问

1.2 参数特性

- ◆ 模组：ESP8266_12F
- ◆ 射频芯片：ESP8266EX
- ◆ 天线形式：板载 PCB 天线
- ◆ 无线网络协议：IEEE 802.11b/g/n
- ◆ 频谱范围：标准 2.4G 频段
- ◆ 发射功率：IEEE 802.11b: 16±2dBm, IEEE 802.11g: 14±2dBm, IEEE 802.11n: 13±2dBm
- ◆ 接收灵敏度：IEEE 802.11b: 11Mbps, IEEE 802.11g: 54Mbps, IEEE 802.11n: HT20(MCS7)
- ◆ 传输距离：宽阔状态下约 50 米，隔墙状态下约 20 米
- ◆ 工作电压范围：3V~5V
- ◆ 工作温度范围：-40°C~85°C
- ◆ 串口速率：最高可达 4Mbps，默认 115200bps
- ◆ 安全性：WEP/WPA-PSK/WPA2-PSK
- ◆ 加密类型：WEP64/ WEP128/TKIP/AES
- ◆ SPI Flash：32Mbit(4MB)
- ◆ 工作模式：STA/AP/STA+AP
- ◆ 二次开发：支持
- ◆ 开发方式：AT 指令开发和 SDK 开发等

第二章 使用说明

2.1 模块说明

（建议一边打开模块原理图并且一边对着实物来看）

模块上有两个 LED 灯，PCB 板载的 LED1 上电后常亮，模组上的 LED 在 GPIO2 低电平时点亮

U2 为 LDO 稳压芯片，将输入电压 3.3V~5V 转为 3.3V

U3 ESP8266_12F 模组

RST: 复位，低电平有效，默认上拉

ADC: ADC 采集，悬空，未用

EN: 芯片使能端，高电平有效，默认上拉

IO16: GPIO16，悬空，未用

IO14: GPIO14，悬空，未用

IO12: GPIO12，悬空，未用

IO13: GPIO13，悬空，未用

VCC: 3.3V 供电

TXD: UART0_TXD/GPIO1，默认上拉

RXD: UART0_RXD/GPIO3，默认上拉

IO5: GPIO5，悬空，未用

IO4: GPIO4，悬空，未用

IO0: GPIO0，默认上拉为运行模式，外部拉低为下载模式

IO2: UART1_TXD/GPIO2，未引出

IO15: GPIO15，默认下拉

GND: 接地

模组的 pin17~pin22，不可用

模式	CH_PD(EN)	RST	GPIO15	GPIO0	GPIO2	TXD0
下载模式	高	高	低	低	高	高
运行模式	高	高	低	高	高	高

模组启动模式说明，部分引脚芯片内部已上拉

2.2 模块接口



选异步通信的串口即可

编号	WIFI 模块	主控
1	VDD	3V3 或 5V
2	GND	GND
3	TXD	任意 UART_RX
4	RXD	任意 UART_TX
5	RST	任意 GPIO_PIN_X
6	IO	悬空或 3V3

注：IO 引脚为 GPIO0，也可用 GPIO 脚控制，接 GND 进入下载模式，进行固件烧录

2.3 程序流程

ESP8266 是串口型 WIFI，速度比较低，不能用来传输图像或者视频这些大容量的数据，主要应用于数据量传输比较少的场合，比如温湿度信息，一些传感器的开关量等

ESP8266 支持 STA 模式、AP 模式和 STA+AP 混合模式

STA 模式，WIFI 模块为连接到无线网络的终端（站点），可以连接到 AP，该模式对应 TCP 传输协议中的客户端（TCP Client）

AP 模式，WIFI 模块产生热点，提供无线接入服务，允许其它无线设备接入，提供数据访问，该模式对应 TCP 传输协议中的服务端（TCP Server）

STA+AP 混合模式，两种模式的共存模式，既能连接其他设备提供的无线网络，又能产生热点，供其他设备连接

TCP/UDP/透传

TCP 是一种面向连接的，提供可靠交付服务和全双工通信的，基于字节流的端到端的传输层通信协议

- 1.TCP 在传输数据之前必须先建立连接，数据传输结束后要释放连接
- 2.每一条 TCP 连接只能有 2 个端点，故 TCP 不提供广播或多播服务
- 3.TCP 提供可靠交付，通过 TCP 连接传输的数据，无差错、不丢失、不重复、并且按序到达
- 4.TCP 是面向字节流的。虽然应用进程和 TCP 的交互是一次一个数据块(大小不等)，但 TCP 把应用程序交下来的数据看成仅仅是一连串的无结构的字节流。TCP 并不知道所传输的字节流的含义

UDP 是一种无连接的，尽最大努力交付的，基于报文的端到端的传输层通信协议

- 1.UDP，在发送数据之前不需要建立连接
- 2.UDP 不保证可靠交付，主机不需要设置复杂的连接状态
- 3.UDP 是面向报文的。UDP 对应用层交下来的报文，既不开并，也不拆分，而是保留这些报文的边界，即应用层交给 UDP 多长的报文，UDP 就照样发送，即一次发送一个报文。在接收端，UDP 一次交付一个完整的报文
- 4.UDP 没有拥塞控制，网络出现的拥塞不会使源主机的发送速率降低
- 5.UDP 支持一对一、一对多、多对一和多对多的交互通信
- 6.UDP 的首部开销小，只有 8 个字节，比 TCP 的 20 个字节的首部要短

透传即透明传输，是指在传输中不管传输的内容是什么，只关心传输时的源地址和目标地址，对数据内容不做任何改变

固件是写入存储器中的程序，在单片机中就是写到 Flash 中的程序

模块可以使用 ESP8266 官方提供的 SDK 进行二次开发，也可使用烧录的 AT 固件，通过 AT 指令配置模块进行使用，这里仅介绍使用 AT 固件，通过串口将 AT 指令传输到 ESP8266 中去

AT 指令可以细分为四种类型

命令形式	AT 指令格式	说明
测试指令	AT+<x>=?	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且参数不可变的功能

注意：

- 1.不是每条 AT 指令都具备上述 4 种类型的命令
- 2.[] 括号内为缺省值，不必填写或者可能不显示
- 3.使用双引号表示字符串数据“string”，例如：AT+CWSAP="ESP8266","1234567890",1,4
- 4.默认波特率为 115200
- 5.AT 指令必须大写，并且以回车换行符结尾“\r\n”

目前 AT 指令集分为旧版 NONOS-AT 和新版 ESP-AT:

NONOS-AT: 使用的 ESP8266 AT 版本, 早期用于 ESP8266 系列芯片, 基于 ESP8266_NONOS_SDK 开发, 已停止更新

ESP-AT: 支持多个芯片平台, 与 NONOS-AT 相比, 支持更多种类的命令, 如 Bluetooth 命令、Bluetooth LE 命令、以太网命令、驱动命令等

新版 ESP-AT 命令集已不支持部分旧指令, 如 AT+CWDHCP 已不支持 AT+CWDHCP_CUR 和 AT+CWDHCP_DEF 加后缀的指令, 使用时请以固件具体支持的命令集为准

介绍一些常用 AT 指令, 更多 AT 指令请查阅参考资料里的 ESP8266 AT 指令集(旧版 NONOS-AT)

基础 AT 指令

AT—测试 AT 启动

执行指令	AT
响应	OK
参数说明	-

AT+RST—重启模块

执行指令	AT+RST
响应	OK
参数说明	-

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

基础 Wi-Fi 功能 AT 指令

AT+CWMODE_CUR—设置当前 Wi-Fi 模式, 不保存到 Flash

指令	测试指令: AT+CWMODE_CUR=?	查询指令: AT+CWMODE_CUR? 功能: 查询 ESP8266 当前 Wi-Fi 模式	设置指令: AT+CWMODE_CUR=<mode> 功能: 设置 ESP8266 当前 Wi-Fi 模式
响应	+CWMODE_CUR:<mode> 取值列表 OK	+CWMODE_CUR:<mode> OK	OK
参数说明	<mode>: • 1: Station 模式 • 2: SoftAP 模式 • 3: SoftAP+Station 模式		
注意	本设置不保存到 Flash		
示例	AT+CWMODE_CUR=3		

AT+CWJAP_CUR—临时连接 AP

指令	查询指令： AT+CWJAP_CUR? 功能：查询 ESP8266 Station 已连接的 AP 信息	设置指令： AT+CWJAP_CUR=<ssid>,<pwd>,[<bssid>],[<pci_en>] 功能：设置 ESP8266 Station 需连接的 AP
响应	+CWJAP_CUR:<ssid>,<bssid>,<channel>,<rssi> OK	OK 或者 +CWJAP_CUR:<error code> FAIL
参数说明	<ul style="list-style-type: none"> • <ssid>: 字符串参数, AP 的 SSID • <bssid>: AP 的 MAC 地址 • <channel>: 信道号 • <rssi>: 信号强度 	<ul style="list-style-type: none"> • <ssid>: 目标 AP 的 SSID • <pwd>: 密码最长 64 字节 ASCII • [<bssid>]: 选填参数, 目标 AP 的 MAC 地址, 一般用于有多个 SSID 相同的 AP 的情况 • [<pci_en>]: 选填参数, 不允许连接 WEP 和 open 的路由器, 可用于 PCI 认证 • <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 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
注意	本设置不保存到 Flash	
示例	AT+CWJAP_CUR="abc","0123456789" 例如, 目标 AP 的 SSID 为 "ab\c", password 为 "0123456789\", 则指令如下: AT+CWJAP_CUR="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_CUR="abc","0123456789","ca:d7:19:d8:a6:44"	

AT+CWSAP_CUR—配置 ESP8266 SoftAP 当前参数

指令	查询指令： AT+CWSAP_CUR? 功能：查询 ESP8266 SoftAP 的配置参数	设置指令： AT+CWSAP_CUR=<ssid>,<pwd>,<chl>,<ecn>[,<max conn>][,<ssid hidden>] 功能：设置 ESP8266 SoftAP 的配置参数
响应	+CWSAP_CUR:<ssid>,<pwd>,<chl>,<ecn>,<max conn>,<ssid hidden>	OK 或 ERROR
参数说明	<ssid>: 字符串参数, 接入点名称 <ul style="list-style-type: none"> • <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 	同左 注意: 指令只有在 SoftAP 模式开启后有效

	<ul style="list-style-type: none"> • [<max conn>] (选填参数)：允许连入 ESP8266 SoftAP 的最多 Station 数目，取值范围 [1, 8] • [<ssid hidden>] (选填参数)：默认为 0，开启广播 ESP8266 SoftAP SSID <ul style="list-style-type: none"> ▸ 0：广播 SSID ▸ 1：不广播 SSID 	
注意	本设置不保存到 Flash	
示例	AT+CWSAP_CUR="ESP8266","1234567890",5,3	

AT+CWDHCP_CUR—设置 DHCP，不保存到 Flash

指令	查询指令： AT+CWDHCP_CUR?	设置指令： AT+CWDHCP_CUR=<mode>,<en> 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: SoftAP DHCP 关闭 ▸ 1: SoftAP DHCP 开启 • Bit1: <ul style="list-style-type: none"> ▸ 0: Station DHCP 关闭 ▸ 1: Station DHCP 开启 	<mode>: <ul style="list-style-type: none"> ▸ 0: 设置 ESP8266 SoftAP ▸ 1: 设置 ESP8266 Station ▸ 2: 设置 ESP8266 SoftAP 和 Station • <en>: <ul style="list-style-type: none"> ▸ 0: 关闭 DHCP ▸ 1: 开启 DHCP
注意	<ul style="list-style-type: none"> • 本设置不保存到 Flash • 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_CUR=0,1	

AT+CIPSTA_CUR—设置 ESP8266 Station 的 IP 地址，不保存到 Flash

指令	查询指令： AT+CIPSTA_CUR? 功能：查询 ESP8266 Station 的 IP 地址	设置指令： AT+CIPSTA_CUR=<ip>[,<gateway>,<netmask>] 功能：设置 ESP8266 Station 的 IP 地址
响应	+CIPSTA_CUR:<ip> +CIPSTA_CUR:<gateway> +CIPSTA_CUR:<netmask> OK	OK
参数说明	注意： ESP8266 Station IP 需连上 AP 后，才可以查询	<ul style="list-style-type: none"> • <ip>：字符串，ESP8266 Station 的 IP 地址 • [<gateway>]：网关 • [<netmask>]：子网掩码
注意	<ul style="list-style-type: none"> • 本设置不保存到 Flash • 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 失效 ▸ 以最后的设置为准 	
示例	AT+CIPSTA_CUR="192.168.6.100","192.168.6.1","255.255.255.0"	

AT+CIPAP_CUR—设置 ESP8266 SoftAP 的 IP 地址，不保存到 Flash

指令	查询指令： AT+CIPAP_CUR? 功能：查询 ESP8266 SoftAP 的 IP 地址	设置指令： AT+CIPAP_CUR=<ip>[,<gateway>,<netmask>] 功能：设置 ESP8266 SoftAP 的 IP 地址
响应	+CIPAP_CUR:<ip> +CIPAP_CUR:<gateway> +CIPAP_CUR:<netmask> OK	OK
参数说明	<ul style="list-style-type: none"> • <ip>：字符串，ESP8266 SoftAP 的 IP 地址 • [<gateway>]：网关 • [<netmask>]：子网掩码 	
注意	<ul style="list-style-type: none"> • 本设置不保存到 Flash • 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 失效 ▸ 以最后的设置为准 	
示例	AT+CIPAP_CUR="192.168.5.1","192.168.5.1","255.255.255.0"	

注：部分 AT 指令后缀为_CUR 的配置不保存到 Flash 内，使用后缀为_DEF 将会将配置保存到 Flash 内

TCP/IP 指令

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 如果连接已经存在，则返回 ALREADY CONNECTED	
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	

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 CONNECTED	
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 • [<UDP local port>]: UDP 传输时, 设置本地端口 • [<UDP mode>]: UDP 传输的属性, 若透传, 则必须为 0 <ul style="list-style-type: none"> ▸ 0: 收到数据后, 不更改远端目标, 默认值为 0 ▸ 1: 收到数据后, 改变一次远端目标 ▸ 2: 收到数据后, 改变远端目标 <p>注意: 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	

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 CONNECTED	
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"> • ESP8266 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。 <p>用户可以使用指令 AT+CIPSSLSIZE=<size> 增大 SSL 缓存</p>	
示例	AT+CIPSSLSIZE=4096 AT+CIPSTART="SSL","iot.espressif.cn",8443	

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
示例	AT+CIPMUX=1 AT+CIPSERVER=1,1001

AT+CPIMODE—设置传输模式

指令	查询指令: AT+CPIMODE? 功能: 查询传输模式	设置指令: AT+CPIMODE=<mode> 功能: 设置传输模式
响应	+CPIMODE:<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, ESP8266 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+CPIMODE=1	

用任意单片机选取任意两个异步通信的 UART 串口, 串口的 RX 和 TX 做数据输入脚和数据输出脚, 任选一个 GPIO 脚做复位控制脚

数据输入脚和数据输出脚分别接到 ESP8266 模块的 TXD 和 RXD 脚

复位控制脚接到 ESP8266 模块的 RST 脚

另一个串口用于 debug 打印调试信息和将接收的数据转发给 ESP8266 模块

1. 选定好 UART 串口, 串口的 RX 与 ESP8266 模块的 TXD 脚相连, 串口的 TX 与 ESP8266 模块的 RXD 脚相连

2. 选定好 GPIO 脚, 该脚与 ESP8266 模块的 RST 脚相连

3. 分别初始化两个串口, 串口的 RX 脚即 RXD 脚, 重点配置为浮空输入模式, 串口的 TX 脚即 TXD 脚, 重点配置为复用推挽输出模式

4.初始化 GPIO 脚即 RST 脚，重点配置为推挽输出模式

5.分别初始化两个 UART，配置工作参数，设置中断优先级，使能串口接收中断

6.分别编写中断服务函数，缓存串口数据保存到数组中，包括数据和长度

7.当串口数据接收完毕，调用库函数将接收的数据用另一个串口再发送一次，如 debug 串口收到串口调试助手发的数据，ESP8266 串口将数据再转发给 ESP8266 模块；ESP8266 串口接收到 ESP8266 模块返回的数据通过 debug 串口打印

8.编写对 ESP8266 模块发送 AT 指令函数

9.发送 AT 测试，若未接收到 OK，控制 RST 脚复位 ESP8266 模块

10.参考 ESP8266 AT 指令使用示例，按需要使用对应的 AT 指令

更多详细内容可参考芯片数据手册、ESP8266 AT 指令集