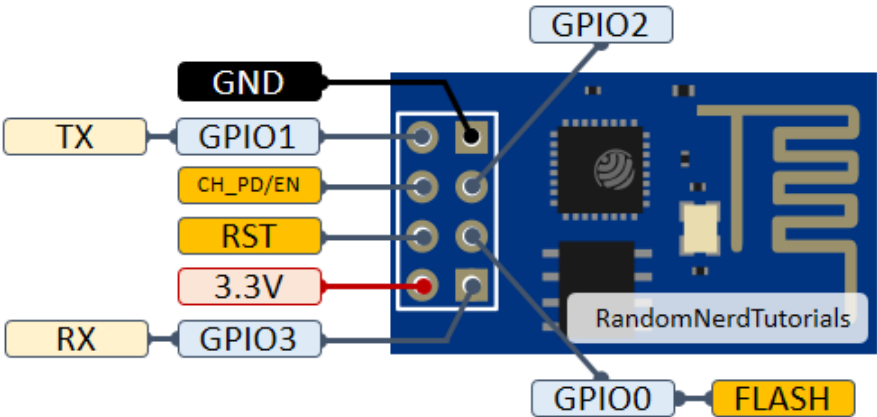
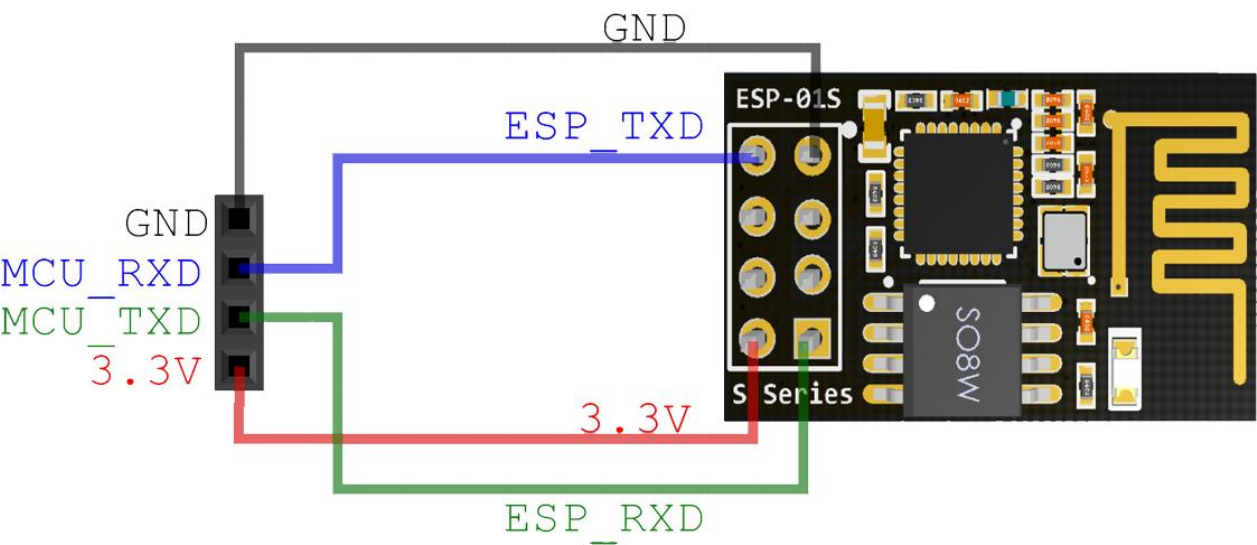


ESP-01S

PIN定义



工作时连线方法



ESP-01S	USB2TTL/MCU	
GND	GND	
TX(GPIO1)	RX	
RX(GPIO3)	TX	
3.3V	3.3V	

相关文件下载

- 固件及烧录软件
<https://pan.baidu.com/s/1bDJ-iltAehvmpxl6ECjm-g> 提取码: t2np
- Putty下载, 下载其中的 a .ZIP archive of all the above 就可以
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

ESP-01S烧录

硬件准备

1. ESP-01S开发板
这个板很小, 输出为2排4列的8pin
2. ESP-01/01S下载器
3. 或ESP-01/01S转USB转接卡
如果没有下载器, 也可以用ESP-01/01S专用的USB转接卡, 在加电前将GPIO0接地(接GND)就能进入烧录模式
4. 或自己接线, 使用TTL2USB的转接卡, 自己连线

ESP-01S进入烧录状态与TTL的连线方式

```
3.3V  -> 3.3V
RST   -> 3.3V
EN     -> 3.3V
GPIO2 -> 3.3V
GND    -> GND
GPIO0  -> GND
TX     -> RX
RX     -> TX
```

软件准备

- Putty 或者 Tera Term 等串口终端软件
- Flash下载工具 flash_download_tools, 乐鑫官网下载地址 <https://www.espressif.com/zh-hans/support/download/other-tools>
 - 当前使用的版本是3.6.4 (flash_download_tools_V3.6.4.zip)
- ESP8266固件, 乐鑫官网下载地址 <https://www.espressif.com/zh-hans/support/download/at> 选择 ESP8266 或 ESP8266 Legacy
 - ESP8266_DOUT_8Mbit_v1.5.4.1.zip 这个是出厂固件, 单bin格式
 - ESP8266_NonOS_AT_Bin_V1.7.4.zip 这个是NonOS AT Bin的最后一个版本, 按分区多bin格式
 - 本文不涉及最新的ESP8266 IDF AT Bin烧录

烧录出厂固件

固件文件 ESP8266_DOUT_8Mbit_v1.5.4.1.zip, 出厂固件只有一个bin文件, 从0x00000开始整个刷写, 与后面升级用的分区写入不一样, 这个要注意.

1. ESP-01S上电进入烧录模式
2. 运行ESPFlashDownloadTool_v3.6.4, 点击ESP8266 DownloadTool按钮
3. 在出现的窗口中, 选择SPIDownload标签页(默认是这一页)
4. 在固件选择栏中, 只需要勾选一行, 其他都不勾选
5. 在勾选的这行
 1. 选择Ai-Thinker_ESP8266_DOUT_8Mbit_v1.5.4.1-a_20171130.bin 文件, 注意文件名中有8Mbit
 2. 地址为 0x00000
6. 检查下面的参数配置: CrystalFreq 26M, SPI SPEED是40MHz, SPI MODE是DIO, FLASH SIZE是8Mbit
7. 点击START
8. 工作正常的话, 右边DETECTED INFO会显示板子信息, 下边进度条栏会显示进度
9. 完成后, 将ESP-01S断电, 再以正常模式上电
10. 打开putty, 连接ESP-01S的COM口, 输入AT+GMR, 按Ctrl+M, Ctrl+J发送命令, 查看输出的版本号是否正确

烧录升级固件

固件文件ESP8266_NonOS_AT_Bin_V1.7.4.zip, 升级固件是按分区提供的, 有多个bin文件, 看起来更复杂一些

1. ESP-01S上电进入烧录模式
2. 运行ESPFlashDownloadTool_v3.6.4, 点击ESP8266 DownloadTool按钮
3. 在出现的窗口中, 选择SPIDownload标签页(默认是这一页)
4. 在固件选择栏中, 勾选需要写入的行, 文件和地址分别为
 1. boot_v1.7.bin 0x00000
 2. user1.1024.new.2.bin 0x01000

3. esp_init_data_default_v08.bin 0xfc000
4. blank.bin 0x7e000
5. blank.bin 0xfe000
5. 检查下面的参数配置: CrystalFreq 26M, SPI SPEED是40MHz, SPI MODE是DIO, FLASH SIZE是8Mbit
6. 界面右下方, 选择正确的COM口, 波特率使用默认的115200
7. 点击START
8. 工作正常的话, 右边DETECTED INFO会显示板子信息, 下边进度条会显示进度
9. 完成后, 将ESP-01S断电, 再以正常模式上电
10. 打开putty, 连接ESP-01S的COM口, 输入AT+GMR, 按Ctrl+M, Ctrl+J发送命令, 查看输出的版本号是否正确

ESP-01S AT 操作

不需要下载各种串口助手软件, 直接用putty就可以了. 如果需要发送16进制的字符, 可以用Tera Term

使用Putty连接和发送命令

putty的使用:

1. 选择serial, 填入COM编号, 例如COM5, 波特率设为115200, 直接连接.
2. 发送命令的操作: 输入命令例如 `AT`, 然后按 `Ctrl+M`, `Ctrl+J` 发送.

关于Ctrl+J的说明

因为一些模块的AT指令需要以 `\r\n` 结束, 这两个分别代表了 `CR` 和 `LF`, 在putty中, 回车时会输出CR, 也可以通过Ctrl+M输出CR, 但是要输出LF, 需要使用Ctrl+J

- Ctrl+M : Carriage Return("`\r`")
- Ctrl+J : Line Feed("`\n`")

如果你需要输出上面的AT指令, 就需要这样进行输入 `A -> T -> Ctrl+M或者Enter -> Ctrl+J`, 我一般用 `Ctrl+M + Ctrl+J`

使用Tera Term发送16进制字符

在命令行中输入16进制并发送是比较麻烦的一件事, 可以先使用可以编辑16进制的文本编辑器, 例如UltraEdit, EmEditor等, 将16进制内容写入文件, 然后在Tera Term中Send File, 勾选Binary就行了.

PS: 在EmEditor中使用Hex View, 菜单入口是File->Reload->Binary

ESP-01S常用命令

先说一下最常用的几个, 如果只是需要连上自己的WIFI看看IP, 用下面的命令组合就可以了

```
# 基础测试, 应该响应OK
AT
# 显示固件版本信息
AT+GMR
# 查看当前的WIFI模式
AT+CWMODE?
# 设置为station模式
AT+CWMODE=1
# 列出2.4G WIFI热点
AT+CWLAP
# 连接热点
AT+CWJAP="SSID", "password"
# 查看IP
AT+CIPSTA?
# ping
AT+PING="202.38.64.5"
```

以下是分组的详细命令列表

基本命令

- AT: 测试, 模块正常应当返回OK
- AT+RST: 重启模块
- AT+GMR: 检查固件版本信息
- ATE: 配置 AT 命令的回显.
 - ATE0: 关闭回显
 - ATE1: 打开回显
- AT+UART_CUR? 查看当前的UART配置
- AT+UART_DEF? 查看默认的UART配置

WIFI命令

- AT+CWMODE?: 查看当前的WIFI模式(Station/SoftAP/Station+SoftAP), 用=号可以设置
 - AT+CWMODE=3
- AT+CWLAP: 列出周围的WIFI AP, 需要先设置为station模式, AT+CWMODE=1
- AT+CWJAP: 连接到WIFI AP, 命令格式 AT+CWJAP="DXQ","aa123456"
 - 掉电重启后, 模块会自动连接之前连接的WIFI
- AT+CWQAP: 断开与WIFI AP的连接
- AT+CIPSTAMAC: 查看客户端模式时的MAC地址
- AT+CIPSTA?: 查看客户端模式的IP地址及掩码
- AT+CIPAPMAC?: Query/Set the MAC address of an ESP SoftAP.
- AT+CIPAP?: Query/Set the IP address of an ESP SoftAP.
- AT+CWSAP?: 查询SoftAP模式下的配置(SSID名称)

TCP/IP命令

- AT+CIPSTATUS: 获取TCP/UDP/SSL连接状态和信息, 先会显示一个状态, 然后会列出各个连接
 - 0: The ESP station is not initialized.
 - 1: The ESP station is initialized, but not started a Wi-Fi connection yet.
 - 2: The ESP station is connected to an AP and its IP address is obtained.
 - 3: The ESP station has created a TCP/SSL transmission.
 - 4: All of the TCP/UDP/SSL connections of the ESP device station are disconnected.
 - 5: The ESP station started a Wi-Fi connection, but was not connected to an AP or disconnected from an AP.
- AT+PING: ping指定的地址, 返回平均响应时间
 - AT+PING="192.168.110.1"
- AT+CIPSTART: 建立TCP/UDP/SSL连接
 - AT+CIPSTART="TCP","192.168.110.1",22
 - AT+CIPSTART="TCPv6","test-ipv6.com",80
 - AT+CIPSTART="UDP","192.168.101.110",1000,1002,2 第4个参数是本地端口, 第5个是模式, UDP unicast
 - AT+CIPSTART="UDPv6","FF02::FC",1000,1002,0 UDP multicast based on IPv6 network
 - AT+CIPSTART="SSL","iot.espressif.cn",8443
- AT+CIPSEND: 透传模式发送数据
- AT+CIPSENDEX: Send data in the normal transmission mode in expanded ways.
- AT+CIPCLOSE: 关闭TCP/UDP/SSL连接
- AT+CIFSR: 获取本地IP地址和MAC地址
- AT+CIPSNTPTIME: 查询SNTP时间
- AT+CIPMUX?: 查询连接类型, 0单连接, 1多连接
 - AT+CIPMUX=1 设置为多连接
- AT+CIPSERVER: 创建或删除一个 TCP/SSL 服务, 创建前需要执行AT+CIPMUX=1
 - AT+CIPSERVER=1,80 创建一个TCP服务
 - AT+CIPSERVER=1,443,"SSL",1 创建一个SSL服务
 - AT+CIPSERVER=0,1 删除一个服务并关闭所有连接
- AT+CIPSERVERMAXCONN?: 查询允许的服务最大连接数量
- AT+CIPSTO?: 查询本地TCP服务超时时间
- AT+CIPSNTPCFG?: 查询 time zone and SNTP server.
- AT+CIPMODE: 查询传输模式
 - 0: 普通传输模式

- 1: Wi-Fi 透传模式, 只有在TCP单连接模式, 以及UDP+remote host and port do not change模式, 或者SSL单连接模式时允许

命令使用技巧

使用ATE0命令关闭回显

在使用上位机通过串口操作ESP8266时, 建议一开始初始化时就将回显关闭.

在使用中, 上位机使用RX中断接收命令响应, ESP8266响应的每个字节都会产生一个中断, 上位机在每次收到中断时, 除了将这个字节记录到rx buffer, 还会设定一个定时器, 例如20毫秒, 这样的效果就是如果归零前持续收到响应, 计时器就持续恢复到20毫秒, 一直到20毫秒内都没有收到响应, 定时器触发中断让 **响应接收完成** 置位, 这时一直阻塞在观察响应接收位的程序将rx buffer中的字符串取出去判断.

如果命令带回显, 因为有些命令需要较长时间响应, 那么在你输入命令时就已经有回显字节, 计时器已经开始工作, 但是在回显结束后, 真正的响应还没那么快回来, 而定时器已经超出了20毫秒的区间, 触发中断让 **响应接收完成** 置位, 程序根据现在的内容去判断响应结果, 就出错了.

关于透传模式的说明

https://docs.espressif.com/projects/esp-at/en/latest/AT_Command_Set/TCP-IP_AT_Commands.html

Enter the data sending mode. AT will send a packet every 20 ms or when the data reaches 2048 bytes. When a single packet containing +++ is received, the ESP device will exit the data sending mode under the Wi-Fi passthrough mode. Please wait for at least one second before sending the next AT command. This command can only be used for single connection in the Wi-Fi passthrough mode. For UDP Wi-Fi passthrough, the parameter has to be 0 when using AT+CIPSTART.

相关的命令

```
# 此时IPMUX必须为0
AT+CIPMUX?

# 开启UDP连接 M+J
AT+CIPSTART="UDP", "192.168.12.210", 3333
# 或者TCP连接 M+J
AT+CIPSTART="TCP", "192.168.12.210", 3333

# 检查IPMODE, 0是普通, 1是透传
AT+CIPMODE?
# 设置为透传, 透传不允许指定发送长度, 此时从远程传入的信息前面不带`+IPD,18:`这样的前缀(18是后面跟着的信息长度)
AT+CIPMODE=1

# IPMODE=1时, 不再接受带长度的`AT+CIPSEND=16`指令, 只能使用不带长度的指令
# 开启本地不受长度限制的传输
AT+CIPSEND
# 此时会出现`>`号, 此时任何输入都会直接发给远程, 而远程的所有发送也会发到本地
# 退出方式: 在20ms内(一个单独的包)输入`+++`这个字符串, 可以直接粘贴.
```

在IPMODE=0的普通传输模式下, 在 **AT+CIPSEND=16** 时, 也可以在 **>** 提示符中使用 **+++** 退出发送

```
AT+CIPSEND=16

OK
>
SEND Canceled
```

注意: **+++** 必须在20ms内(一个单独的包)输入完毕, 必须是完整的三个+号, 不能多也不能少, 不能和前面或者后面的命令连接, 否则都会无效.

ESP-01S测试

准备工作

1. 一个2.4GHz的WIFI AP
2. 连接到这个AP的一个linux机器(需要运行nc和tcpdump)
3. 带USB连接的ESP-01S, 连到工作电脑

测试步骤

1. 在Linux机器启动nc和tcpdump

```
# 开启udp监听
nc -u -l 192.168.17.81 3333
# 同时开启tcpdump监听
tcpdump -i eth0 udp port 3333 -vv -n
```

2. ESP-01S连接到电脑

在putty连接到对应的COM口, 执行命令

```
# 检查版本
AT+GMR
AT version:1.7.4.0 (May 11 2020 19:13:04)
SDK version:3.0.4 (9532ceb)
compile time:May 27 2020 10:12:17
Bin version (Wroom 02):1.7.4
OK

# 切换到station模式
AT+CWMODE=1

OK

# 查看可用的wifi列表
AT+CWLAP
+CWLAP: (4, "NSKY-Guest", -66, "e0:1c:41:30:69:55", 1, 41, 0, 5, 3, 7, 0)
+CWLAP: (4, "NSKY-ceshi", -66, "e0:1c:41:30:69:56", 1, 41, 0, 5, 3, 7, 0)
+CWLAP: (4, "NSKY", -67, "e0:1c:41:30:67:94", 1, 48, 0, 5, 3, 7, 0)
+CWLAP: (4, "NSKY-Guest", -67, "e0:1c:41:30:67:95", 1, 50, 0, 5, 3, 7, 0)
+CWLAP: (4, "NSKY-ceshi", -69, "e0:1c:41:30:67:96", 1, 48, 0, 5, 3, 7, 0)
+CWLAP: (4, "MDM-Development", -73, "88:25:93:67:bf:a9", 1, 53, 0, 4, 4, 7, 0)
+CWLAP: (4, "NSKY", -74, "e0:1c:41:30:68:54", 1, 53, 0, 5, 3, 7, 0)
+CWLAP: (3, "DIRECT-b2-HP M427 LaserJet", -52, "6a:14:01:62:3c:b2", 4, 58, 0, 4, 4, 7, 1)

OK

# 加入WIFI
AT+CWLAP="NWF", "123.1234"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

OK

# 查看获得的IP
AT+CIPSTA?
+CIPSTA:ip:"192.168.17.182"
+CIPSTA:gateway:"192.168.17.1"
+CIPSTA:netmask:"255.255.255.0"
```

OK

注意: 当ESP-01S发起UDP连接时, linux机器的监听不管是否开启, 只要ESP-01S发送了数据, tcpdump都能接收到数据

```
# 发起UDP连接
AT+CIPSTART="UDP","192.168.17.81",3333,3333,2
CONNECT

OK

# 启用数据发送, 设置长度为16
# 返回OK后, 进入发送数据模式, 可以在键盘输入字符, 当输入\0字符或输入长度超过16都会触发发送动作, 发送完之后会
回到AT交互
AT+CIPSEND=16

OK
>
Recv 16 bytes

SEND OK
```

3. 这时候在linux机器上, 可以在nc界面看到接收到的信息, 在tcpdump界面可以看到监测到的udp包

nc

```
# nc -u -l 192.168.17.81 3333
this is a test w
```

tcpdump

```
[root@vm_n01 ~]# tcpdump -i eth0 udp port 3333 -vv -nn
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:00:34.787356 IP (tos 0x0, ttl 128, id 65, offset 0, flags [none], proto UDP (17), length 44)
    192.168.17.182.dec-notes > 192.168.17.81.dec-notes: [udp sum ok] UDP, length 16
```

通过STM32操作ESP-01S

使用STM32F103C8T6

连线方式

ESP-01S	STM32	USB2TTL
GND	GND	GND
TX(GPIO1)	PA10	
RX(GPIO3)	PA9	
3.3V	3.3V	
	PA3	TX
	PA2	RX

代码地址: <https://github.com/IOsetting/stm32f103-esp8266>

使用说明:

- 创建一个一级目录MDK-ARM, 使用Keil MDK5在这个目录下创建项目
- 在项目group设置中, 包含对应的C文件
- 在项目C/C++设置中, 包含对应的头文件目录
- 在user目录下, 创建config.h, 加入

```
#ifndef _CONFIG_H
#define _CONFIG_H

const char WIFI_SSID[] = "WIFI_SSID";
const char WIFI_PASSWORD[] = "password";

#endif
```

使用STM32F401CCU6

连线方式

ESP-01S	STM32	USB2TTL
GND	GND	GND
TX(GPIO1)	PA3	
RX(GPIO3)	PA2	
3.3V	3.3V	
	PA10(RX)	TX
	PA9(TX)	RX
	PA12(RX)	
	PA11(TX)	

代码地址 <https://github.com/IOsetting/stm32f401-esp8266>

其他与F103C8T6相同

参考

- 乐鑫官网下载地址 <https://www.espressif.com/zh-hans/support/download/at>
- AT英文站 <https://docs.espressif.com/projects/esp-at/en/latest/index.html>
- ESP8266的一系列技术文章,值得推荐 <https://blog.csdn.net/xh870189248/article/details/77985541>