# ESP8266 Embedded Wi-Fi Serial Communication Module

# AT command
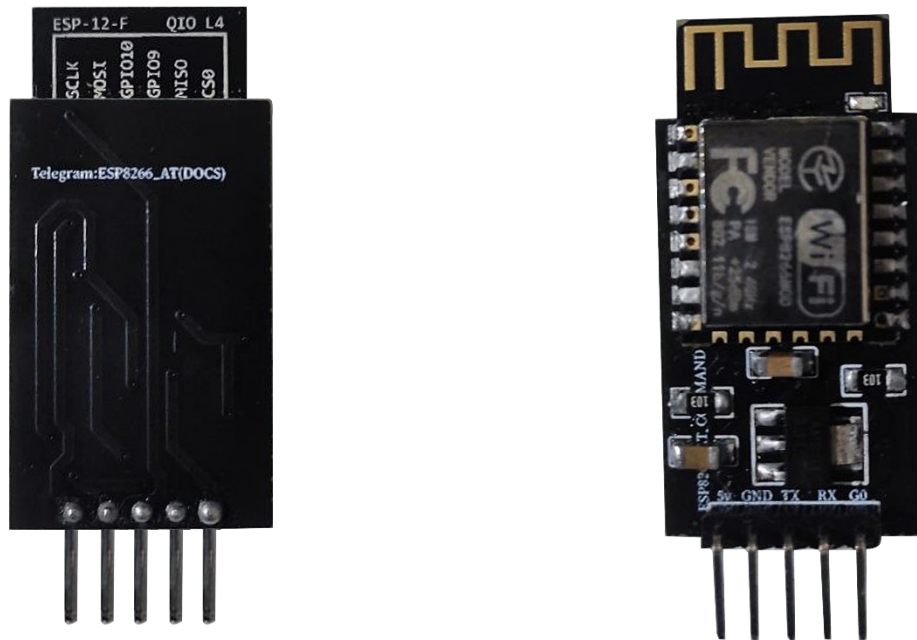
| Pins | | |
|---|---|---|
| 5v | Connect to 5v power | |
| GND | GND | |
| TX | Connect it to the RX of MCU | UART |
| RX | Connect it to the TX of MCU | |
| G0 | NC | |

Note: when you powered the module make sure that G0 pin not connected to anything

Please visit the channel to see all update and upgrade

Developed by Yaman jaddouh

**Commands list**

| WICON | WISTA | WIIP | BAUD |
|---|---|---|---|
| HTGET | HTPOST | VERSION | RESET |
| DEFUALT | CHECKUPDATE | UPDATE | FSREAD |
| FSREMOVE | FSLIST | FSADD | FSWRITE |
| MACADDR | NOWINIT | NOWDEINIT | NOWADD |
| NOWSEND | | | |

All command above should be written with this way:

***AT+<COMMAND>***

all commends are case-sensitive so don't use small case with it**,**

**Command Details**

When you want to deals and comminutes with the module you need to know which commands you should use.

Here are the commands:

1. Test the module.

    test the serial connection is Ok

    | Command | Response | Parameters |
    |---|---|---|
    | **AT** | OK | None |

2. Reset the module.
    reboot the module

    | Command | Response | Parameters |
    |---|---|---|
    | **AT+RESET** | OK | None |

3. Reset setting to default

    | Command | Response | Parameters |
    |---|---|---|
    | **AT+DEFUALT** | OK | None |

The default data are: SSID and PASSWORD are empty and the   baud rate: 9600 bps

4. Check if there is an update or note

| Command | Response | Parameters |
|---|---|---|
| **AT+CHECKUPDATE** | If not connected to Wi-Fi<br>**WIFI NOT CONNECTED**<br>If there is an update<br>**THERE IS UPDATE AVIALBLE: \<version\>**<br>Else<br>**THERE IS NO UPDATE AVIALBLE** | None |

5. Update the firmware

| Command | Response | Parameters |
|---|---|---|
| **AT+UPDATE** | If not connected to Wi-Fi<br>**WIFI NOT CONNECTED**<br>If there is an update<br>It will start downloading new Firewarm immediately<br>Else<br>**THERE IS NO UPDATE AVIALBLE** | None |

When start updating it will appear the percent of progress, please don't

Shutdown the module while updating

6. Wi-Fi Connection

Connect the module to Wi-Fi use the following command as below.

| Read command<br>**AT+WICON?** | Response<br>- **\<SSID\>,\<PASS\>** when Wi-Fi is connected |
|---|---|

| | |
|---|---|
| | - **NO WIFI CONNECTED** when Wi-Fi is not connected |
| | Parameters<br>**None** |
| | Note<br>**None** |
| | Response<br>**OK** |
| Write command<br>**AT+WICON=<SSID>, <PASS>** | Parameters<br>**SSID**: the name of the Wi-Fi network wants to connect.<br>**PASS**: the password of SSID.<br><br>When they left empty it will be disconnected from Wi-Fi network |
| | Note<br>- It takes several times for connecting.<br>- When the module is not connecting to the Wi-Fi, it will blink for 0.1 sec<br>- When the module is connecting to the Wi-Fi, it will blink for 2 sec |

7. Wi-Fi IP Address

   When the module is connected to the Wi-Fi, it has an IP address

| | |
|---|---|
| Read Command<br>**AT+WIFIP?** | Response<br>**<IP Address>** |
| | Parameters<br>**None** |
| | Note<br>**None** |

8. Baud rate Serial communication

| | |
|---|---|
| Read Command<br>**AT+BAUD?** | Response<br>**<Baud rate>** |
| | Parameters<br>**None** |
| | Note |

| | None |
|---|---|
| write Command<br>**AT+BAUD=< NUMBER >** | Response<br>**OK** |
| | Parameters<br>**NUMBER**<br>1 for 9600 bps<br>2 for 38400 bps<br>3 for 115200 bps |
| | Note<br>- Be careful when changing the speed because you will not be able to communicate with module until you change the speed with your program |

## 9. Version of firmware

| | Response<br>**ESP8266 ATCOMMAND <VERSION>** |
|---|---|
| Read Command<br>**AT+VERSION?** | Parameters<br>**None** |
| | Note<br>**None** |

## 10. GET http request

You can make http request with get method

| | Response<br>**<HTTP CODE>, <HTTP RESPONSE>** |
|---|---|
| Write Command<br>**AT+HTGET=<URL>** | Parameters<br>**URL:** the site or URL you want to make http request for |
| | Note<br>- To use this command, you have to connect to Wi-Fi which has an internet access |

## 11. POST http request

You can make http request with POST method

| | |
|---|---|
| | Response<br>**<HTTP CODE>, <HTTP RESPONSE>** |
| Write Command<br>**AT+HTPOST=<URL>, <DATA>** | Parameters<br>**URL:** the site or URL you want to make http request for<br>**DATA**: the data you want to send as string |
| | Note<br>- To use this command, you have to connect to Wi-Fi which has an internet access |

## 12. List the files.

There are files in memory as SPIFFS or LittleFS.

| | |
|---|---|
| | Response<br><br>if there any file:<br>**<name File>, <size in byte>**<br>else:<br>**NO FILE EXIST** |
| Read Command<br>**AT+FSLIST?** | Parameters<br>**None** |
| | Note<br>- You will get the sequence of response depend on count of exists files |

## 13. Read data in files

| | |
|---|---|
| Write Command<br>**AT+FSREAD=<FILE NAME>** | Response<br>If the file is existing you will get<br>**< file content>**<br>Else YOU WILL GET<br>**ERROR NOT FOUND** |

| | Parameters<br>**FILE NAME:** full name of file with extension |
|---|---|
| | Note<br>**None** |

## 14. Write data into files

| | Response<br>**OK** |
|---|---|
| | Parameters<br>**FILE NAME:** full name of file with extension<br>**FILE CONETNET:** the content of file you want to be saved |
| Write Command<br>**AT+FSWRITE=<FILE NAME>, <FILE CONETNET>** | Note<br>- In writing method if the file doesn't exist, it will create the file and store the data<br>- In writing method if the file exists, it will overwrite the data so the old data will be deleted and will be replaced with the new **<FILE CONTENT>** |

## 15. add data into files

| | Response<br>**OK** |
|---|---|
| | Parameters<br>**FILE NAME:** full name of file with extension<br>**FILE CONETNET:** the content of file you want to be saved |
| Write Command<br>**AT+FSADD=<FILE NAME>, <FILE CONETNET>** | Note<br>- In adding method if the file doesn't exist, it will create the file and store the data<br>- In adding method if the file is exist it will append the data so the old |

| | data won't be deleted and will be replaced with the new **<FILE CONTENT>** |
| --- | --- |

16.remove files

| | Response<br>If file exists<br>**OK**<br>else<br>**ERROR NOT FOUND** |
| --- | --- |
| Write Command<br>**AT+FSREMOVE=<FILE NAME>** | Parameters<br>**FILE NAME:** full name of file with extension<br>- if you pass keyword **ALL** it will delete all the files exist in the memory |
| | Note<br>- In adding method if the file doesn't exist, it will create the file and store the data<br>- In adding method if the file exists it will append the data so the old data won't be deleted and will be replaced with the new **<FILE CONTENT>** |

17.Get the MAC address of the device

| Command | Response | Parameters |
| --- | --- | --- |
| **AT+MACADDR?** | Return MAC address with this formula<br>xx:xx:xx:xx:xx:xx | None |

18.ESP-NOW initialize

| Command | Response | Parameters |
| --- | --- | --- |
| **AT+NOWINIT** | if it initializes without error<br>**Init ESP-NOW done**<br>Else | None |

| | Error initializing ESP-NOW | |
|---|---|---|

This command will begin the [ESP-NOW](#) protocol

19. ESP-NOW DE initialize

| Command | Response | Parameters |
|---|---|---|
| AT+NOWDEINIT | **OK** | None |

20. Add MAC address of device for make a connection with ESP-NOW

| **AT+NOWADD=<MAC address>** | Response<br>If the adding is done without error<br>**OK**<br>else<br>**Failed to add peer** |
| | Parameters<br>**MAC address:** the Mac address to connect with ESP-NOW<br>- |
| | Note<br>**None** |

21. Send data over ESP-NOW protocol

| **AT+NOWSEND=<DATA>** | Response<br>If the message has arrived it will return<br>**OK**<br>else<br>**ERROR RECEIVE** |
| | Parameters<br>**MAC address:** the Mac address to connect with ESP-NOW<br>- |
| | Note |

| | |
|---|---|
| | - When the device receives the data, it will send the received data direct to serial with this formula **NOWDATA=<DATA>**<br>- When the device trying to connect to Wi-Fi, it will affect to ESP-NOW connection until Wi-Fi connection stabilizes |