

Fady Fares

WiFi Module - ESP8266-12

# Description

 The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that’s just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

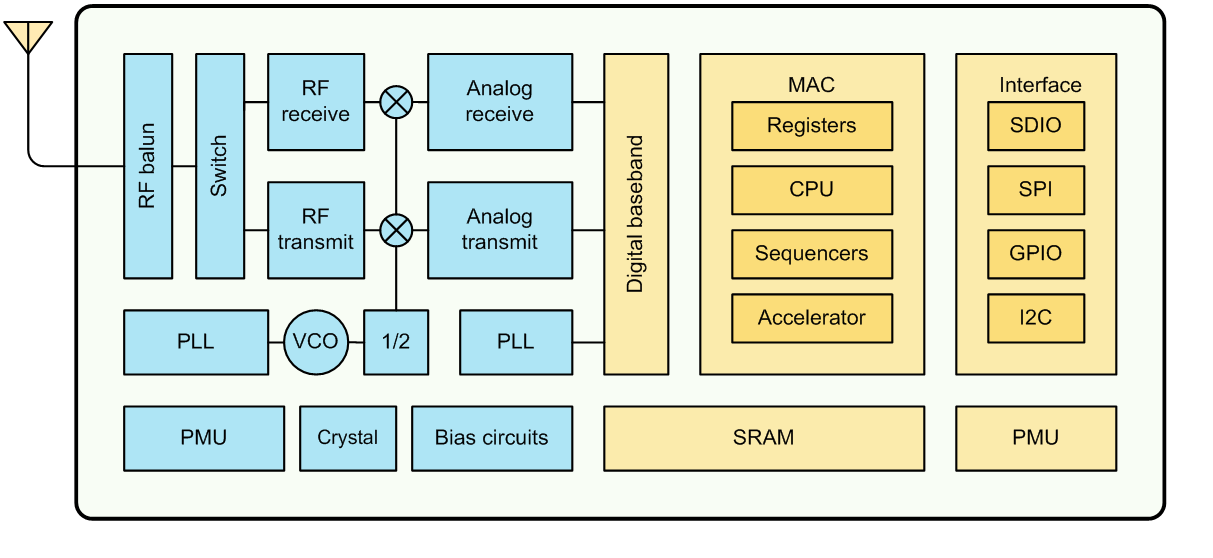
This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

Figure :ESP8266EX Block Diagram

# features

* support wireless 802.11 b/g/n standard
* support the STA/AP/STA + AP three work modes
* support rich Socket the AT command
* support UART/GPIO data communication interface
* support Smart Link intelligent networking
* support the STA/AP/STA + AP three work modes
* built-in 32-bit MCU and can be applied as processor
* low power consumption, suitable for battery-powered applications
* Single 3.3 V power supply

# Pins out

Figure :pins out

# pins connection

|  |  |
| --- | --- |
| **Microcontroller(Atmega32,TivaC,Arduino UNO)** | **ESP8266-12** |
| 3.3v (external power supply is recommended) | VCC & EN & GPIO2 & GPIO0 |
| GND | GND & GPIO15 |
| TX | PIN(16) RX |
| RX | PIN(17) TX |

Table : pins connection

# AT Commands

|  |  |  |
| --- | --- | --- |
| ***Function*** | ***AT Command*** | ***Response*** |
| Working | AT | OK |
| Restart | AT+RST | OK [System Ready, Ve[ndor:w](http://www.ai-thinker.com/)ww[.ai-thinker.com](http://www.ai-thinker.com/)] |
| Firmware version | AT+GMR | AT+GMR 0018000902 OK |
| List Access Points | AT+CWLAP | AT+CWLAP +CWLAP:(4,"RochefortSurLac",- 38,"70:62:b8:6f:6d:58",1)  +CWLAP:(4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1) OK |
| Join Access Point | AT+CWJAP? AT+CWJAP="SSID","Password" | Query AT+CWJAP? +CWJAP:"RochefortSurLac" OK |
| Quit Access Point | AT+CWQAP=? AT+CWQAP | Query OK |
| Get IP Address | AT+CIFSR | AT+CIFSR 192.168.0.105 OK |
| Set Parameters of Access Point | AT+ CWSAP?  AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn> | Query ssid, pwd  chl = channel, ecn = encryption |
| WiFi Mode | AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3 | Query STA AP BOTH |
| Set up TCP or UDP connection | AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART =  <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART=  <id><type>,<addr>, <port> | Query  id = 0-4, type = TCP/UDP, addr = IP address, port= port |
| TCP/UDP  Connections | AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1 | Query Single Multiple |
| Check join devices' IP | AT+CWLIF |  |
| TCP/IP Connection Status | AT+CIPSTATUS | AT+CIPSTATUS? no this fun |
| Send TCP/IP data | (CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length> |  |
| Close TCP / UDP connection | AT+CIPCLOSE=<id> or AT+CIPCLOSE |  |
| Set as server | AT+ CIPSERVER= <mode>[,<port>] | mode 0 to close server mode; mode 1 to open; port = port |
| Set the server timeout | AT+CIPSTO? AT+CIPSTO=<time> | Query  <time>0~28800 in seconds |
| Baud Rate\* | AT+CIOBAUD?  Supported: 9600, 19200, 38400, 74880,  115200, 230400, 460800, 921600 | Query AT+CIOBAUD? +CIOBAUD:9600 OK |
| Check IP address | AT+CIFSR | AT+CIFSR 192.168.0.106 OK |
| Firmware Upgrade (from Cloud) | AT+CIUPDATE | 1. +CIPUPDATE:1 found server  2. +CIPUPDATE:2 connect server  3. +CIPUPDATE:3 got edition  4. +CIPUPDATE:4 start update |
| Received data | +IPD | (CIPMUX=0): + IPD, <len>: (CIPMUX=1): + IPD, <id>, <len>: <data> |
| Watchdog Enable\* | AT+CSYSWDTENABLE | Watchdog, auto restart when program errors occur: enable |
| Watchdog Disable\* | AT+CSYSWDTDISABLE | Watchdog, auto restart when program errors occur: disable |