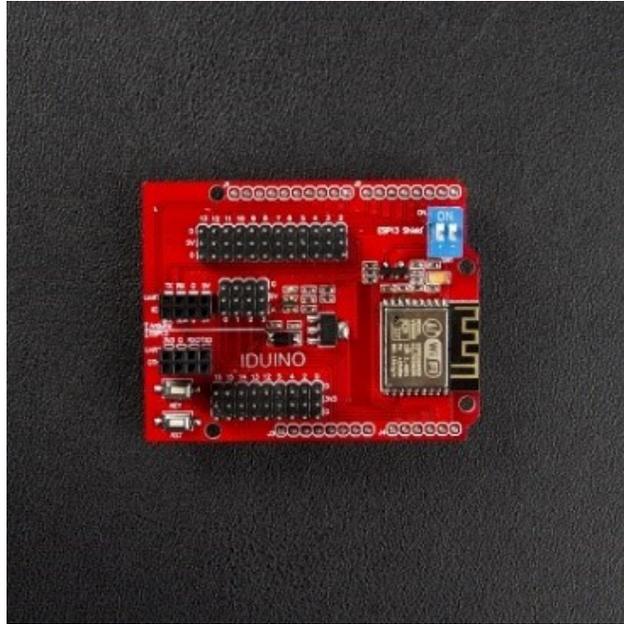


User Manual

Of

ESP8266 Wifi Shield(AD010)



Description

The shield is designed based on esp8266 by Espressif Systems, pin-compatible with Arduino UNO/Mega2560 DevBoard. Which can be used as two independent part, the upper part is for the Arduino UNO/Mega pin-extension board, the bottom part is for the Esp8266 expansion board. Now the Arduino IDE which is higher than 1.6.4 can be installed Esp8266 board package, so you can control the esp8266 board through the Arduino IDE, not just AT Command. The install method please refer the weblink at last of this guide.

The serial wifi shield has the following features:

1. WiFi module is industrial-grade chips ESP8266, which is ESP-12E with metal shield, strong anti-interference ability;
- 2, Shield is pin-compatible with Arduino Uno, Mega2560, and other control board. A voltage converter chip is used to deal with 3.3V (Esp8266) and 5V (Arduino);
- 3, Dual DIP switches is used for serial ports so that this module shield can be used alone as an Arduino Uno expansion board, and also be used as ESP8266 expansion board;
- 4, Serial data is transported to WiFi device transparently, and vice versa. Arduino program does not need any configuration;
- 5, WebServer is developed to configure WiFi parameters and serial port parameters;
- 6, The module shield can be used as an independent ESP8266 development board. for instance, downloading the official AT commands firmware, NodeMCU open source firmware can be used;
- 7, The module shield also can be used as stand-alone expansion board for Arduino Uno.

Technical Specifications

- 802.11 b / g / n wireless standards;
- STA / AP modes support ;
- TCP / IP protocol stack, One socket;
- Supports standard TCP / UDP Server and Client;
- Supports serial port baud rate configuration:
1200/2400/4800/9600/19200/38400/57600/74800/115200 bps;
- Supports serial data bits: 5/6/7/8 bits;
- Supports serial parity: none;
- Supports serial stop bits: 1/2 bit;
- Pin-compatible with Arduino UNO, Mega;
- Arduino Pinout 2/3/4/5/6/7/8/9/10/11/12/13;
- ESP8266 GPIO Pinout 0/2/4/5/9/10/12/13/14/15/16 / ADC / EN / * UART TX / UART RX;
- KEY button: modes configuration;
- Dual-Ports DIP switches: switching Arduino and ESP8266;
- WiFi operation current: continuous transmission operation: $\approx 70\text{mA}$ (200mA MAX), idle mode: $<200\mu\text{A}$;
- Serial WiFi transmission rate: 110-460800bps;
- Temperature: $-40^{\circ}\text{C} \sim +125^{\circ}\text{C}$;
- Humidity: 10%-90% non-condensing;
- Weight: about 20g (0.7oz);

Pinout

This serial WiFi module board provides an interface as shown in PCB view. And each pin functions are silk-printed on the circuit board, as follows:

Part	Pin	Function	Remark
Arduino	G	Ground	
	D	Digital IO	
	A	Analog IO	
	5V	5V	
	TX	Arduino Uno TX	Connection with ESP8266
	RX	Arduino Uno RX	Connection with ESP8266
	SCL	Arduino SCL	
	SDA	Arduino SDA	
ESP8266	G	Ground	
	D	Digital IO	
	3V3	3.3V	
	RX0	ESP8266 RX0	Connection with Arduino
	TX0	ESP8266 TX0	Connection with Arduino
	EN	ESP8266 EN	
	AD	ESP8266 AD	
	RST	ESP8266 reset	
	KEY	ESP8266 WiFi configuration	
	SW	ESP8266 And Arduino	When programming Arduino with IDE, SW1 must switch to "OFF" position.

LED definition

ESP8266-Based serial WiFi Shield has two LEDs. Red color LED is donated to the power supply. Blue LED is a multi-functional LED, and its function is described as follows:

Blink 4 times continuously	Data communication
Blink 1 time every 0.5s	Configuration modes
Blink 1 time every 1s	Connecting router in STA mode
Steady-on	AP/STA working mode

Example:

This wifi shield allows you operate an Esp8266 controller board through Arduino IDE, without any wire connection, just plug this shield into Arduino UNO/Mega board, upload the wifi configuration code to start a wifi host or hotspot. Then use the TCP/UDP Debugging software to monitor your data transmission.

By default, we burn the v0.9.2.3 AT Firmware.bin, and the example in the library compatible with this version firmware. If you need other firmware, please refer the ESP8266 official web:

http://www.esp8266.com/wiki/doku.php?id=loading_firmware

First, the hardware connection is very simple and no wire connection needed, just plug the WIFI shield on the Arduino UNO/Mega board.



Then connect the UNO board to your computer, run the Arduino IDE. Before using the example, you need to install ESP8266 package to your Arduino IDE, you can refer the following step:

<https://github.com/esp8266/Arduino>

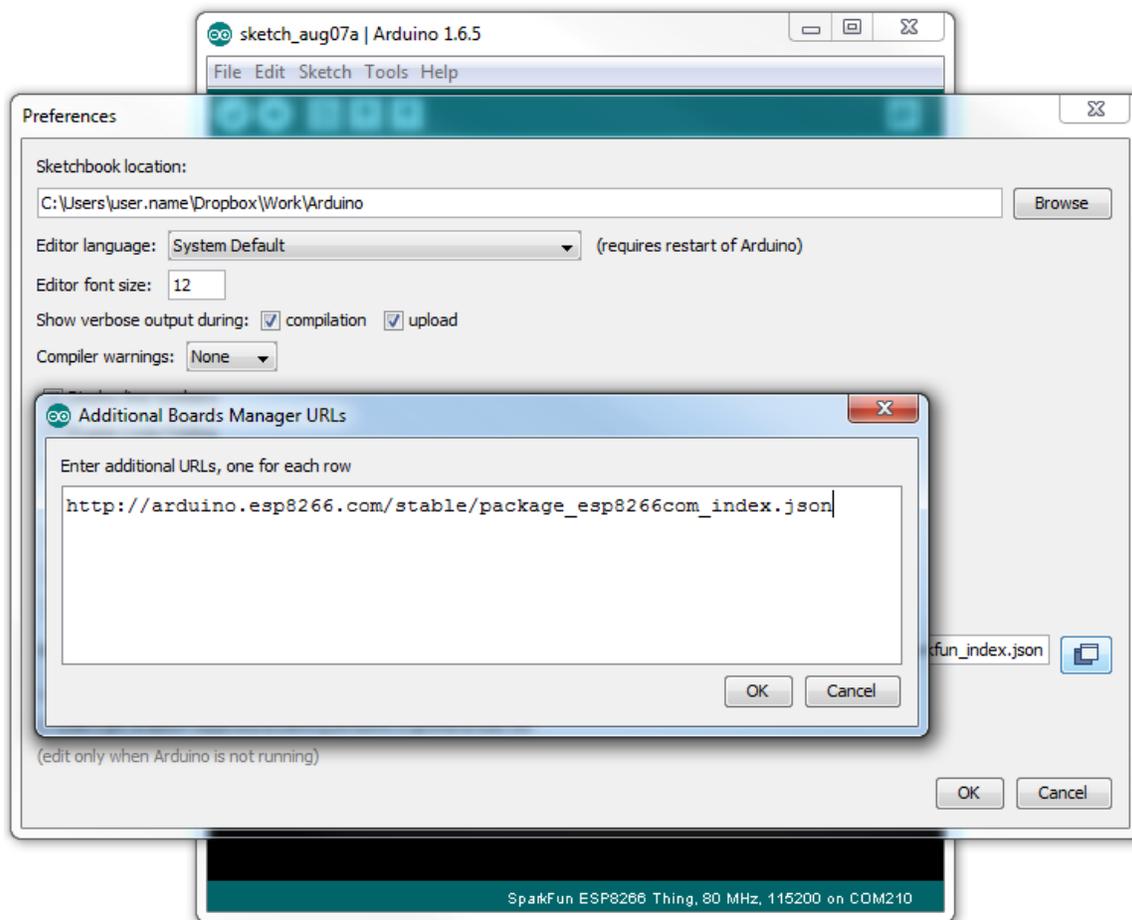
<https://learn.sparkfun.com/tutorials/esp8266-thing-development-board-hookup-guide/setting-up-arduino>

With the release of Arduino 1.6.4, adding third party boards to the Arduino IDE is easily achieved through the board manager. If you're running an older version of Arduino (1.6.3 or earlier), we recommend upgrading now. As always, you can download the latest version of Arduino from arduino.cc.

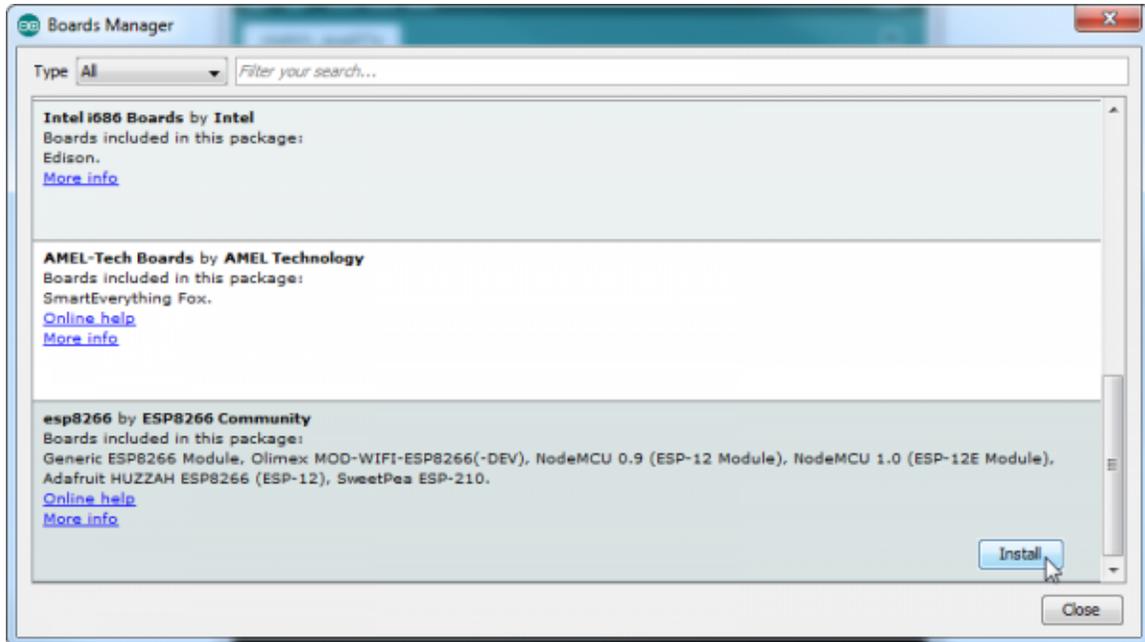
To begin, you'll need to point the Arduino IDE board manager to a custom URL. Open up Arduino, then go to the Preferences (**File > Preferences**). Then, towards the bottom of the window, paste this URL into the "Additional Board Manager URLs" text box:

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

You can add multiple URLs by clicking the window icon, and pasting in one URL per line.



Hit OK. Then navigate to the Board Manager by going to **Tools > Boards > Boards Manager**. Look for **esp8266**. Click on that entry, then select **Install**.



The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, an Arduino-blue "INSTALLED" will appear next to the entry.

After installing the ESP8266 package to your Arduino IDE, you can run the included example to test.

chatServer	2016/5/27 11:48	文件夹	
udpSendReceivingString	2016/5/27 11:48	文件夹	
webclientRepeating	2016/5/27 11:48	文件夹	
webserver	2016/5/27 11:48	文件夹	
wifiTest	2016/12/29 18:02	文件夹	
Read me.txt	2016/3/24 14:36	文本文档	1 KB

Now, we choose the wifi Test code, the example allows the wifi shield connect with your local wifi. Of course, you need to change your local wifi name and password in the example code.

```
wifiTest | Arduino 1.6.9
File Edit Sketch Tools Help

wifiTest

*/

#define SSID      "youlocalwifi"
#define PASSWORD  "12345678"

#include "uartWIFI.h"
#include <SoftwareSerial.h>
WIFI wifi;

Done compiling.

Sketch uses 10,368 bytes (32%) of program storage space. Maximum is 32,256 bytes.
Global variables use 813 bytes (39%) of dynamic memory, leaving 1,235 bytes free.

51 Arduino Uno on COM10
```

Please note when you upload the code to the arduino board, the connection between Arduino RX/TX pin and Wifi shield RX/TX pin must be off. That means you need to turn off the switch (blue button on the wifi shield) while the uploading, when the upload was done, please turn to "on" place.

After this, check your local router, you will find a new "unknown" wifi, which is your wifi shield.

	已连接: 1小时31分57秒	IP地址: 192.168.31.194	MAC地址: 68:3E:34:EE:1D:88	<input type="checkbox"/>
	android-9c893dfa1d2aef7 已连接: 1小时32分15秒	IP地址: 192.168.31.234	MAC地址: FC:DB:B3:1E:8E:B0	<input checked="" type="checkbox"/>
	iPhone-8 已连接: 1小时32分15秒	IP地址: 192.168.31.107	MAC地址: 38:CA:DA:AF:23:8D	<input checked="" type="checkbox"/>
	This is the wifi shield Unknown 已连接: 21分30秒	IP地址: 192.168.31.204	MAC地址: 18:FE:34:DF:48:82	<input checked="" type="checkbox"/>
	CHAIR-MAX 已连接: 1小时32分15秒	IP地址: 192.168.31.221	MAC地址: 54:E4:3A:57:94:87	<input checked="" type="checkbox"/>

That means the connection is successful.

We offer 5 examples to the user, each example has an explanation at the begin of the code. But usually user this board through different firmware, so we introduce another firmware the show how this shield work.

How to burn firmware to the ESP8266 chip

First, we need a flasher tool, the official recommend the NodeMCU flasher,

<https://github.com/nodemcu/nodemcu-flasher>

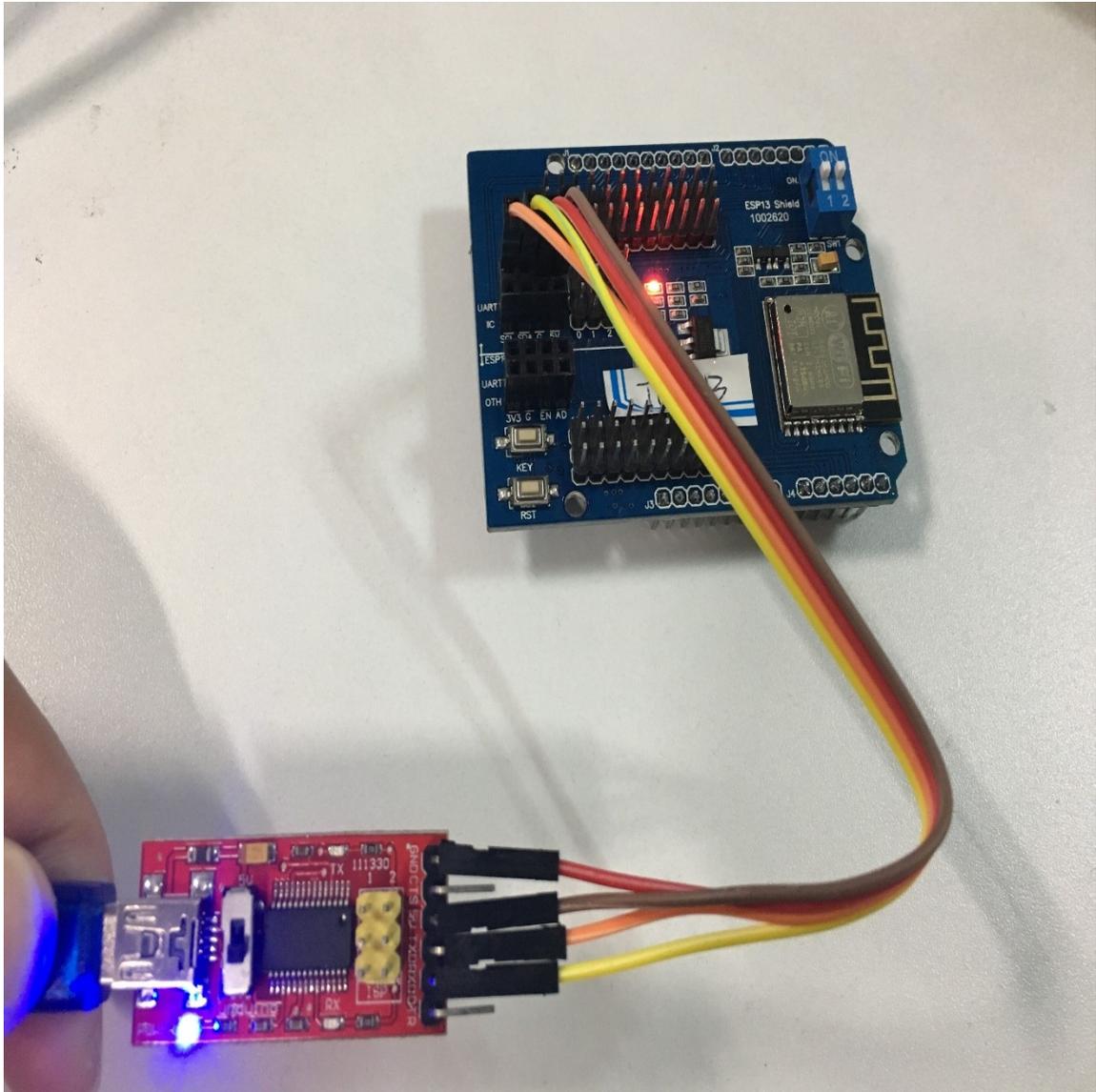
The hardware connection as below, we need a USB to TTL downloader (ST1125),

```

5V-----5V
Gnd-----Gnd
RX-----RX
TX-----TX

```

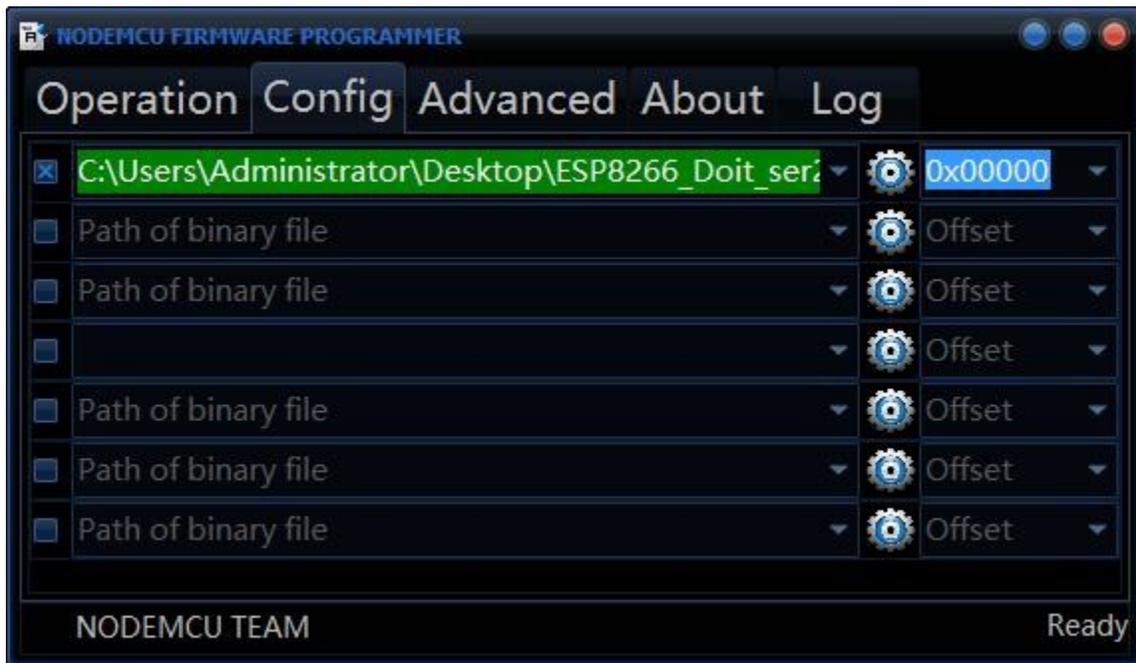
Note that “RX” to “RX” , “TX” to “TX” .



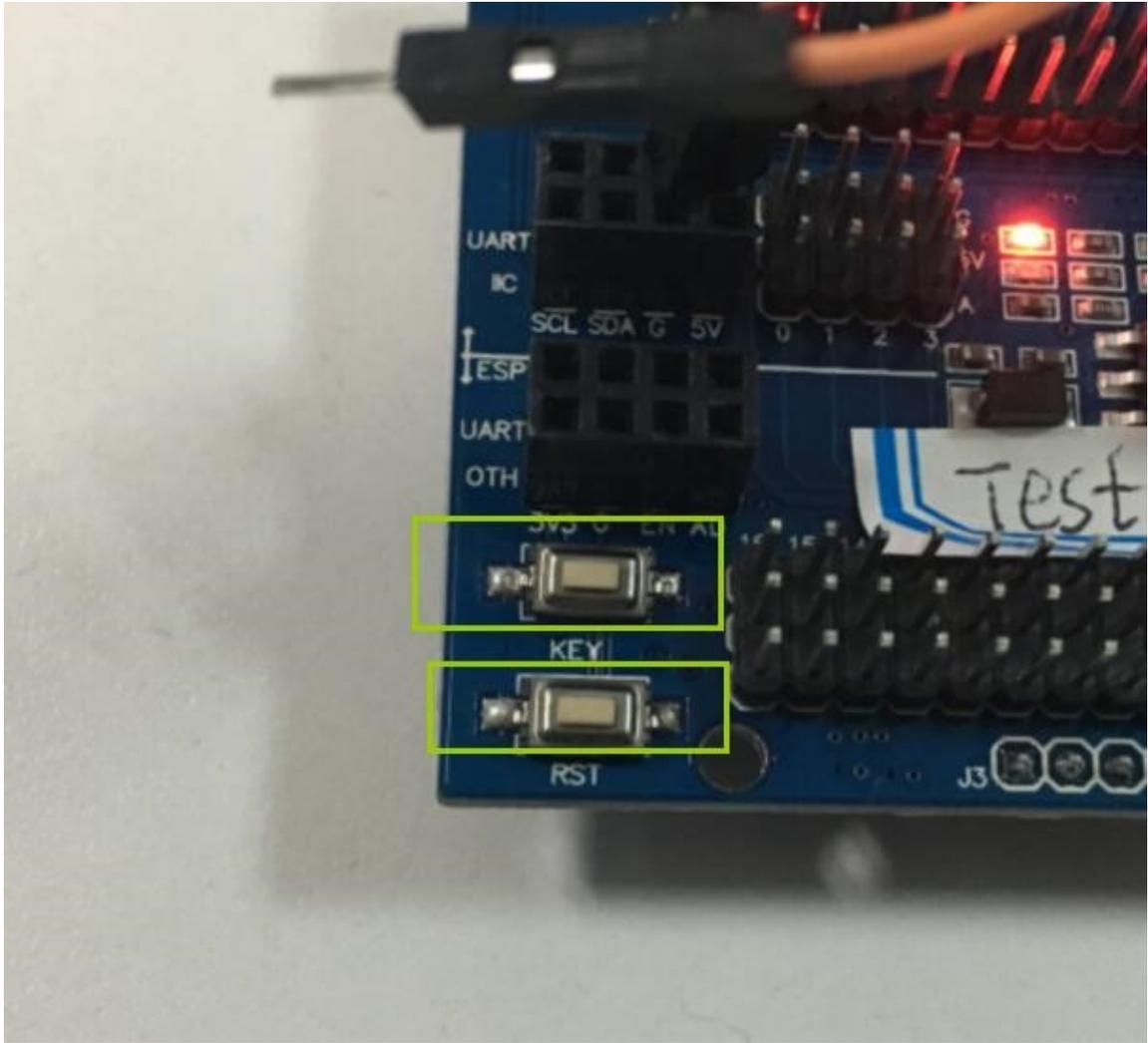
Then run the NodeMCU flasher,

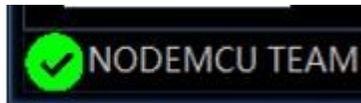
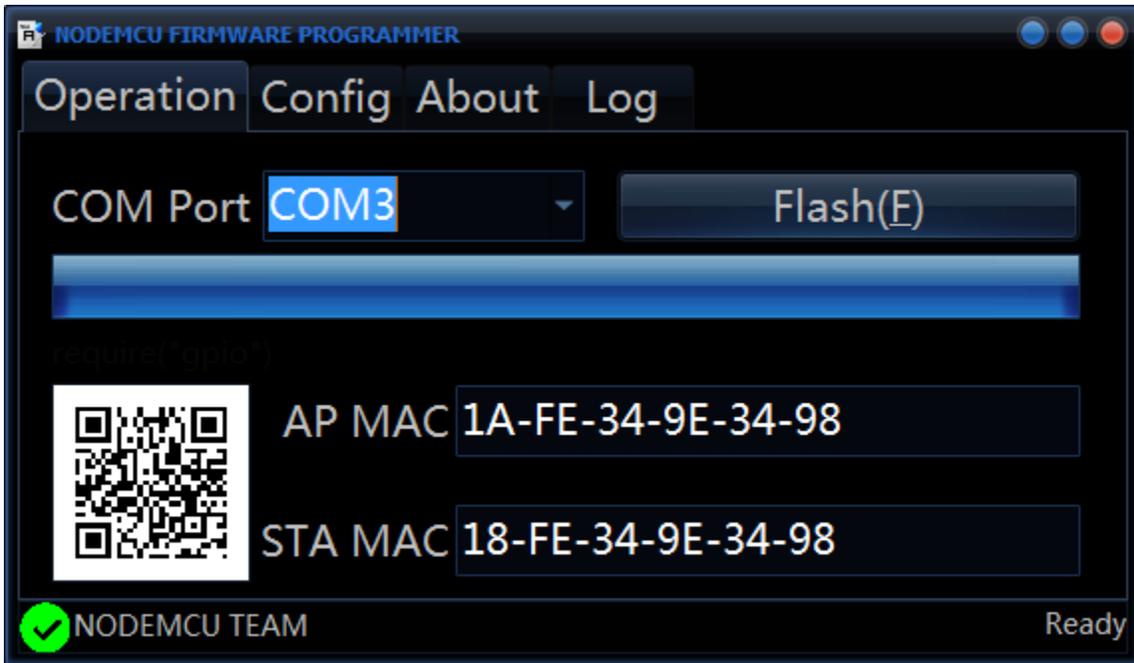


Click the “config” to choose your own firmware,



Here we use doit(www.doit.am) firmware, then click the “Flash” , and press the “key” button first, Meanwhile, press the “ Reset ” button for a few seconds and release, until the progress bar begins to move.





When the above
successfully.

appears, that means the firmware is install

Then plug the wifi shield into the arduino uno and connect with the computer, run the Arduino IDE, also install the esp8266 package.

Now, you need a notebook to connect your wifi shield and receive data.



Then choose your notebook' s wifi, default is Doit-Wifi,



Open your browser, and type the ip" 192.168.4.1" and change wifi access point to your custom name.

type the ip

ESP8266 Serial WiFi Shield

Serial Setting:

Baud : 9600

Databits: 8

Parity: NONE

Stopbits: 1

Access Point(AP) :

change your wifi name

AP name: your wifi name

AP Password: 12345678

Encrypt Method: OPEN

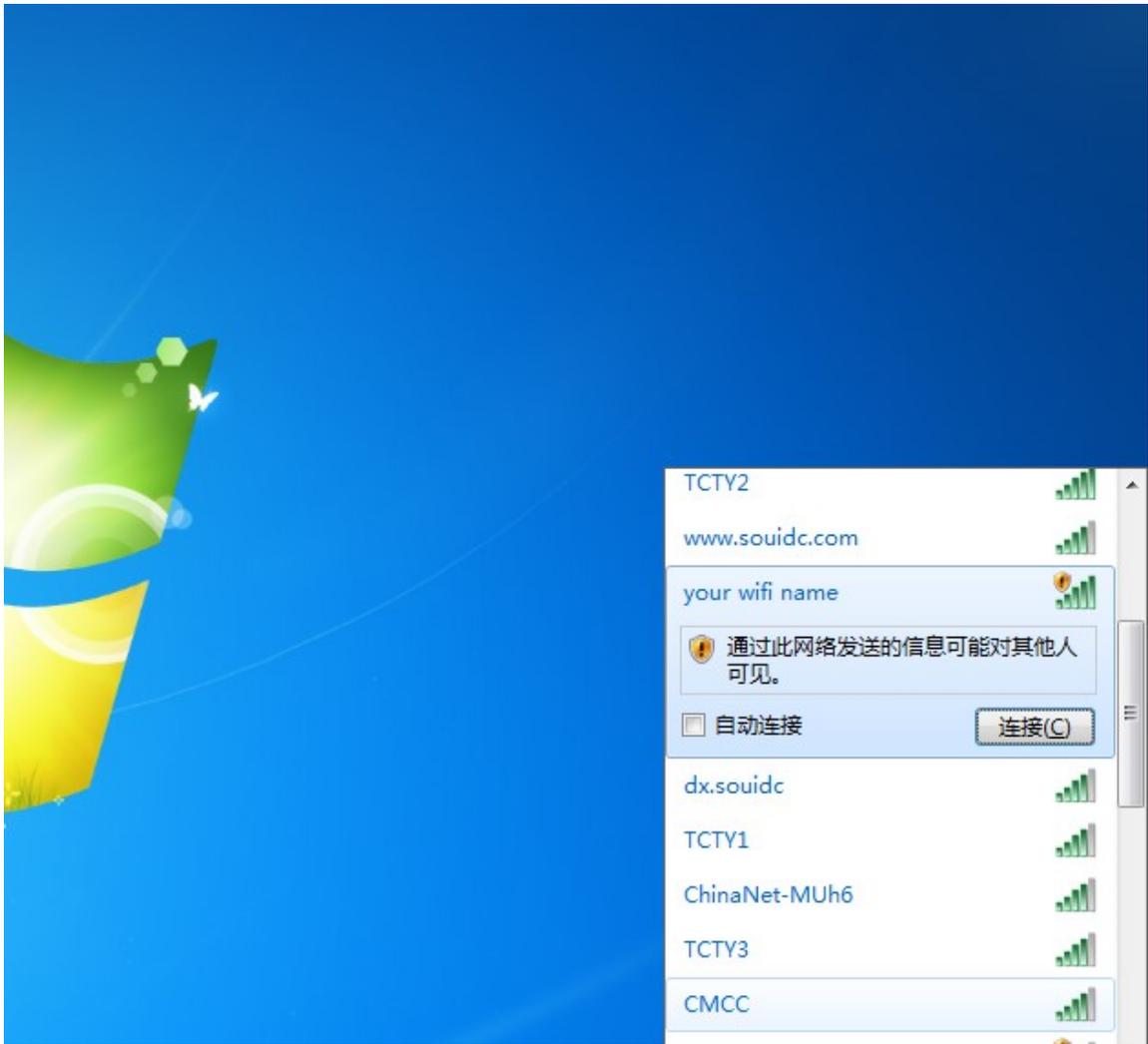
Hide AP: Yes No

AP IP address: 192.168.4.1

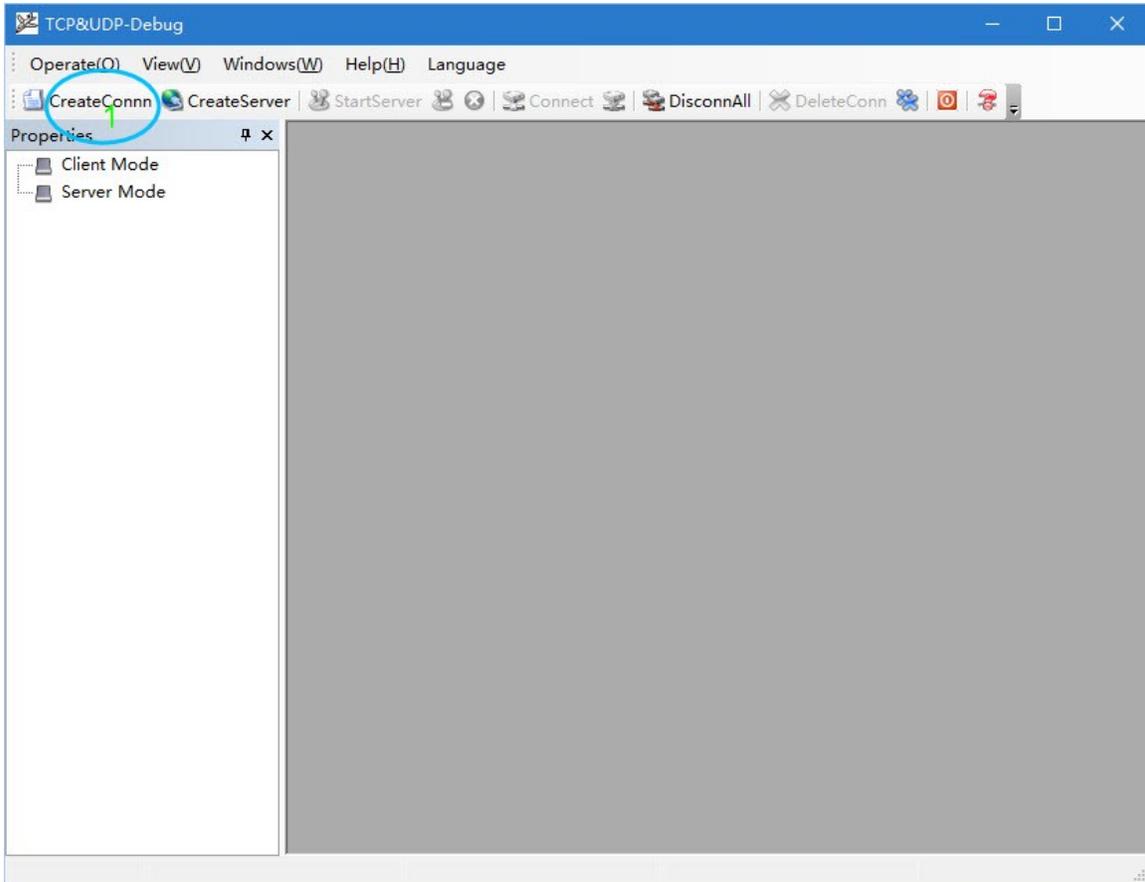
AP Netmask: 255.255.255.0

AP Gateway address: 192.168.4.1

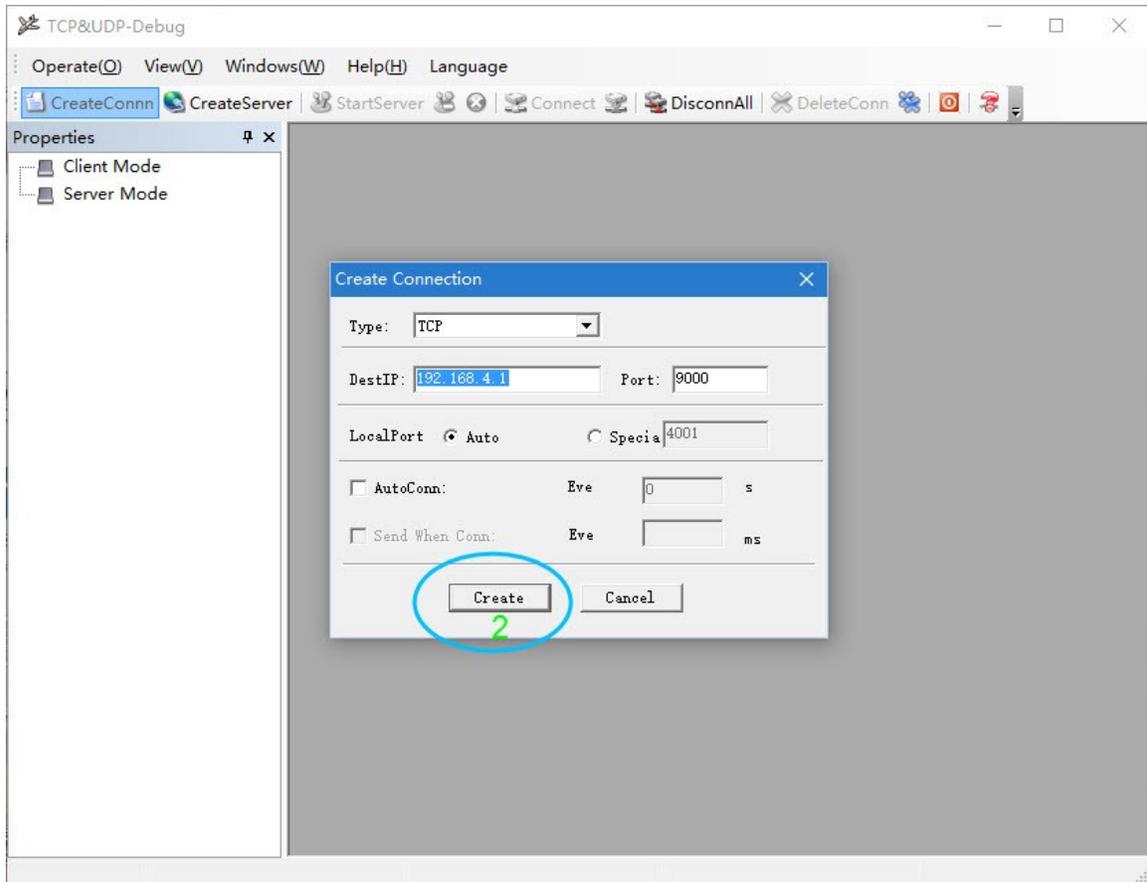
Then click the submit to save the configuration. After few seconds, the wifi shield will reboot and refresh the wifi list of your notebook, you will find your wifi.



Connect your wifi, then the notebook can receive the data from the wifi shield through arduino. But how to receive the data, here we need TCP debugger tool, we recommend the [TCPUDPDbg](#) tool.

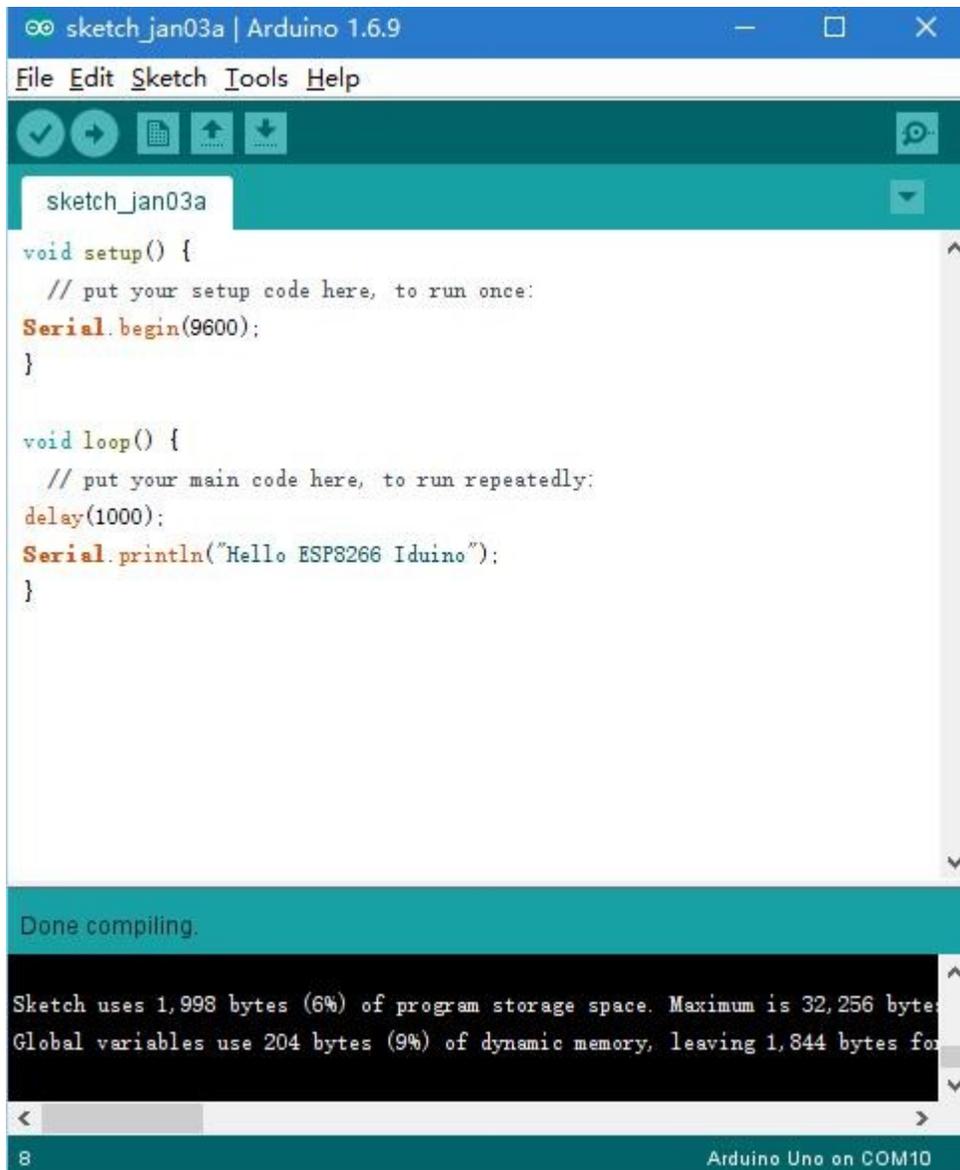


Run the TCP/UDG debugger, click the "CreatConnection" ,



Type the ip "192.168.4.1" , and port" 9000" , then click the "Create" .

Now run the Arduino IDE,



Type the code:

```
*****code begin*****
```

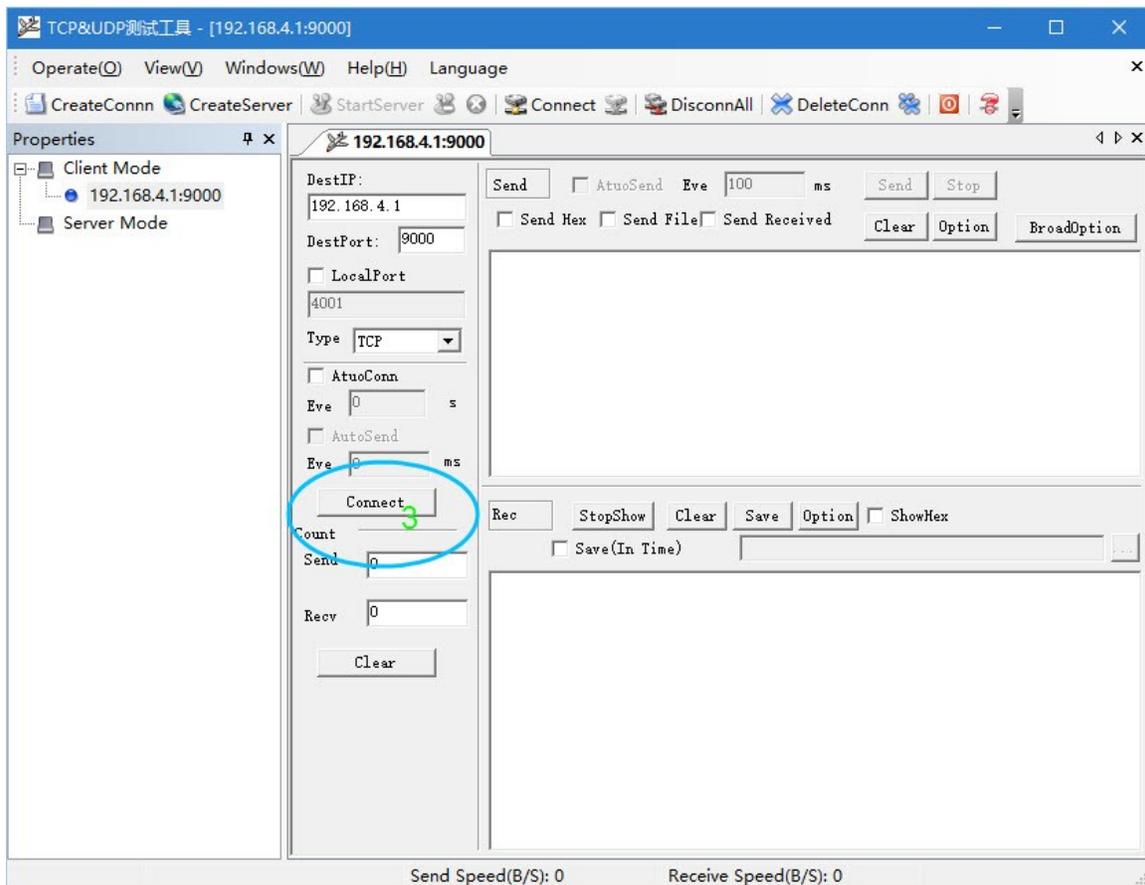
```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}
```

```
void loop() {  
    // put your main code here, to run repeatedly:  
    delay(1000);  
    Serial.println("Hello ESP8266 Iduino");  
}
```

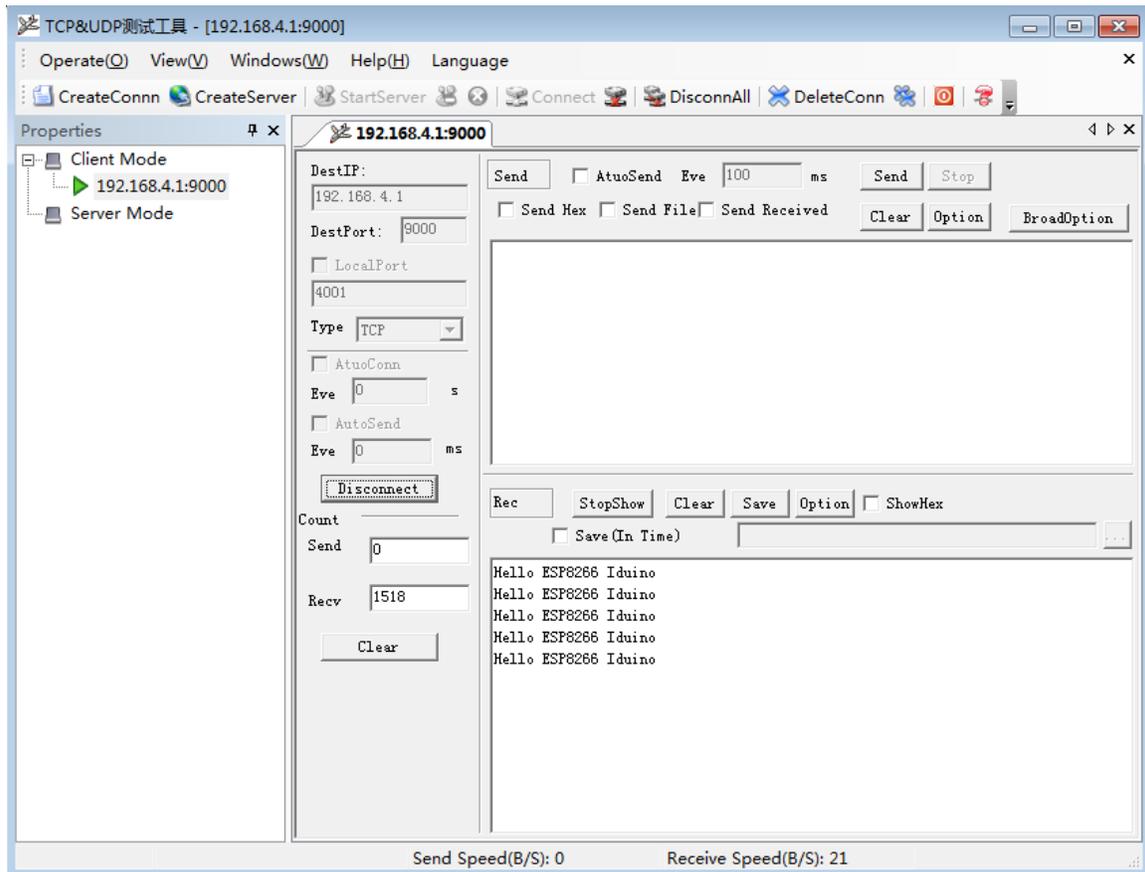
```
*****code End*****
```

This code sent word "Hello ESP8266 Iduino" to your notebook through Arduino UNO and wifi shield.

Now run the TCP/UDG debugger to see whether the notebook receives the data.



Choose the connection "192.168.4.1" and click the "Connect", you will see the word "Hello ESP8266 Iduino" have been received normally,



Now the communication between the wifi shield and your notebook successfully.

Usually, we control ESP8266 through AT Command, with the help of arduino IDE, we can send any AT Command by the above method in a visual window.

Please refer the following information:

<https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide>

<https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/downloads>