

# User Manual--SUSTech-STM32-ESP8266

Group members:

JIAXI ZHANG

SHUN FAN

ZUNYAO MAO

SHILONG LI

Date: 2020/12/22

## Introduction

---

This report is for course CS301( *Embedded System and Microcomputer Principle* ) term project of Wi-Fi.

ATK-ESP8266 is a UART-Wi-Fi module with high performance. This project uses ATK-ESP8266 module to realize the wireless communication between MCU. We establish communication between two ESP8266 Wi-Fi module based on MiniSTM32 board.

## Orders of Instructions

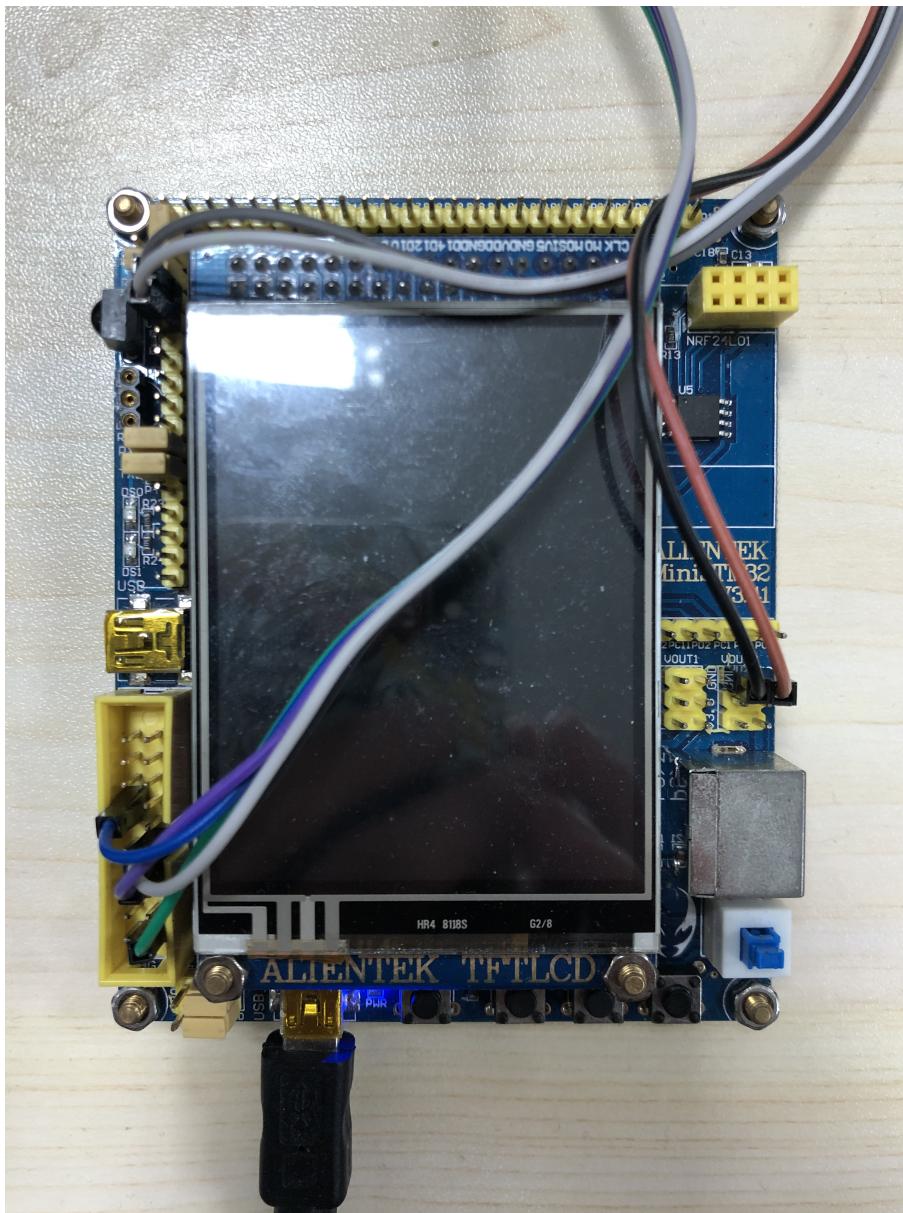
---

- Physical hardware setting up
- Software setting up
  - DMA
  - GPIO
  - NVIC
  - RCC
  - SYS
  - TIM3
  - USART
- Configure the ESP8266 working mode
- Establish default connection
- Communicate with each other
- Disconnection
- Check connection
- LED display

## Hardware connection

As the picture shown below, the hardware connection is set as follows.

	Device port	STM32 port
Wi-Fi ESP8266	VCC	5V
Wi-Fi ESP8266	GND	GND
Wi-Fi ESP8266	TXD	PA3
Wi-Fi ESP8266	RXD	PA2

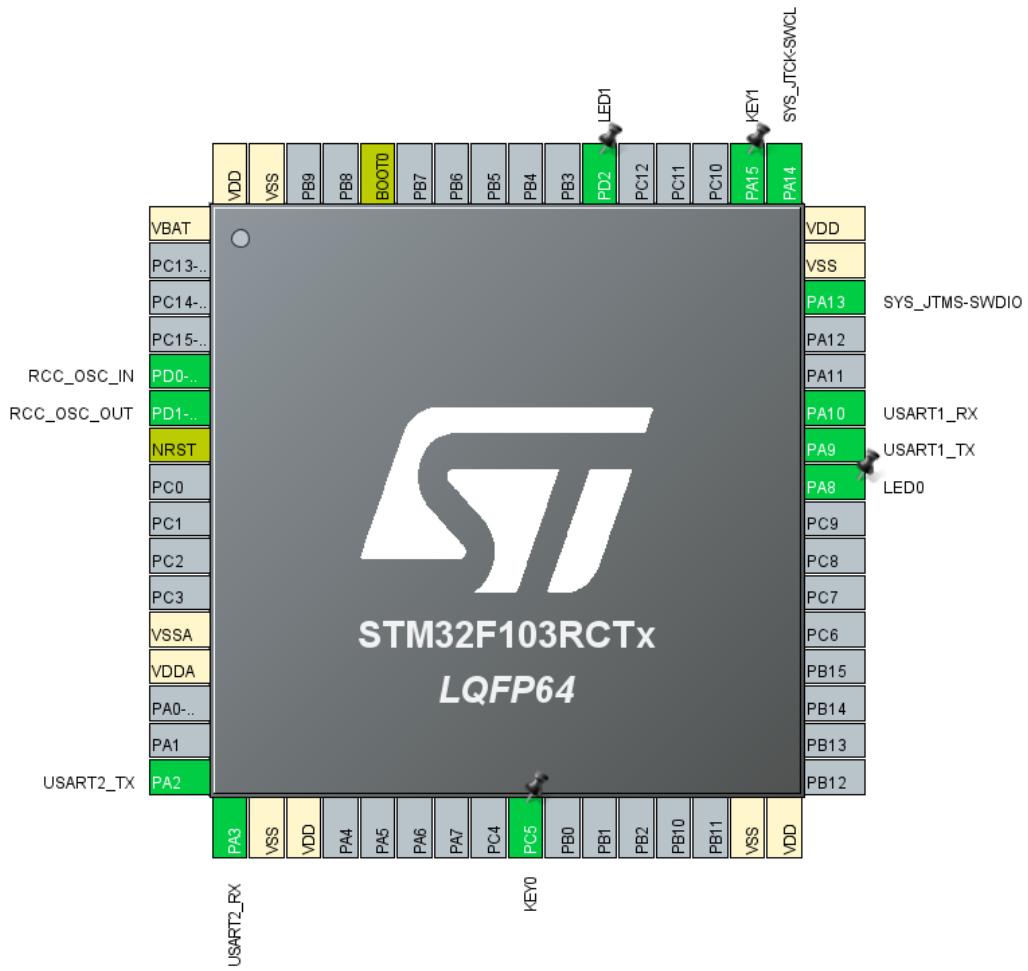


## ATTENTION

1. **VCC** needs to be connected to 5V rather than 3.3V, otherwise, it may encounter voltage shaking.

## Software settings

The pinout view in the STM32CubeIDE is set as follows.



And details are listed below.

DMA

Categories	A-Z	Configuration			
		DMA1	DMA2	MemToMem	
		DMA Request	Channel	Direction	Priority
System Core	▼				
DMA	◆				
GPIO					
IWDG					
NVIC					
RCC	▲				
SYS	▲				
WWDG					

GPIO

Categories A-Z

System Core

- DMA
- GPIO**
- IWDG
- NVIC
- RCC
- SYS
- WWDG

Analog >

Configuration

Group By Peripherals

GPIO  RCC  SYS  USART  NVIC

Search Signals  Search (Ctrl+F)  Show only Modified Pins

Pin...	Signal ...	GPIO ...	GPIO ...	GPIO ...	Maxim...	User L...	Modified
PA8	n/a	Low	Output...	No pull...	Low	LED0	<input checked="" type="checkbox"/>
PA15	n/a	n/a	Extern...	Pull-up	n/a	KEY1	<input checked="" type="checkbox"/>
PC5	n/a	n/a	Extern...	Pull-up	n/a	KEY0	<input checked="" type="checkbox"/>
PD2	n/a	Low	Output...	No pull...	Low	LED1	<input checked="" type="checkbox"/>

Categories A-Z

System Core

- DMA
- GPIO**
- IWDG
- NVIC
- RCC
- SYS
- WWDG

Configuration

Group By Peripherals

GPIO  RCC  SYS  USART  NVIC

NVIC Interrupt Table | Enabled | Preemption Priority | Sub Priority

EXTI line[9:5] interrupts	<input checked="" type="checkbox"/>	2	0
EXTI line[15:10] interrupts	<input checked="" type="checkbox"/>	2	0

## NVIC

Categories A-Z

System Core

- DMA
- GPIO
- IWDG
- NVIC**
- RCC
- SYS
- WWDG

Analog >

Timers >

Connectivity >

CAN

I2C1

I2C2

SDIO

SPI1

SPI2

SPI3

UART4

UART5

Configuration

NVIC  Code generation

Priority Group 4 bits for pre-emption  Sort by Preemption Priority and Sub Priority

Search  Search...    Show only enabled interrupts  Force DMA channels Interrupts

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Prefetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
DMA1 channel5 global interrupt	<input checked="" type="checkbox"/>	0	0
DMA1 channel6 global interrupt	<input checked="" type="checkbox"/>	0	0
EXTI line[9:5] interrupts	<input checked="" type="checkbox"/>	2	0
TIM3 global interrupt	<input checked="" type="checkbox"/>	3	0
USART1 global interrupt	<input checked="" type="checkbox"/>	1	0
USART2 global interrupt	<input checked="" type="checkbox"/>	1	0
EXTI line[15:10] interrupts	<input checked="" type="checkbox"/>	2	0

## RCC

RCC Mode and Configuration

Mode

High Speed Clock (HSE) Crystal/Ceramic Resonator

Low Speed Clock (LSE) Disable

Master Clock Output

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

System Parameters

VDD voltage (V)	3.3 V
Prefetch Buffer	Enabled
Flash Latency(WS)	2 WS (3 CPU cycle)

RCC Parameters

HSI Calibration Value	16
HSE Startup Timeout Value (ms)	100
LSE Startup Timeout Value (ms)	5000

## SYS

SYS Mode and Configuration

Mode

Debug Serial Wire

System Wake-Up

Timebase Source SysTick

Categories A-Z

System Core

- DMA
- GPIO
- IWDG
- NVIC
- RCC
- SYS**
- WWDG

## TIM3

TIM3 Mode and Configuration

**Mode**

Slave Mode	Disable
Trigger Source	Disable
Clock Source	Internal Clock
Channel1	Disable
Channel2	Disable
Channel3	Disable
Channel4	Disable
Combined Channels	Disable

Use ETR as Clearing Source

XOR activation

One Pulse Mode

**Configuration**

Reset Configuration

Parameter Settings    User Constants    NVIC Settings    DMA Settings

Configure the below parameters :

Search (Ctrl+F)    

**Counter Settings**

Prescaler (PSC - 16 bits value)	7199
Counter Mode	Up
Counter Period (AutoReload Register - 16 ... 1999)	1999
Internal Clock Division (CKD)	No Division
auto-reload preload	Enable

**Trigger Output (TRGO) Parameters**

## USART

USART1 Mode and Configuration

**Mode**

Mode	Asynchronous
Hardware Flow Control (RS232)	Disable

**Configuration**

Reset Configuration

User Constants    NVIC Settings    DMA Settings    GPIO Settings

Parameter Settings

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
DMA1 channel5 global interrupt	<input checked="" type="checkbox"/>	0	0
USART1 global interrupt	<input checked="" type="checkbox"/>	1	0

User Constants    NVIC Settings    DMA Settings    GPIO Settings

Parameter Settings

DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 5	Peripheral To Memory	Low

USART2 Mode and Configuration

**Mode**

Mode	Asynchronous
Hardware Flow Control (RS232)	Disable

**Configuration**

Reset Configuration

User Constants    NVIC Settings    DMA Settings    GPIO Settings

Parameter Settings

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
DMA1 channel6 global interrupt	<input checked="" type="checkbox"/>	0	0
USART2 global interrupt	<input checked="" type="checkbox"/>	1	0

User Constants    NVIC Settings    DMA Settings    GPIO Settings

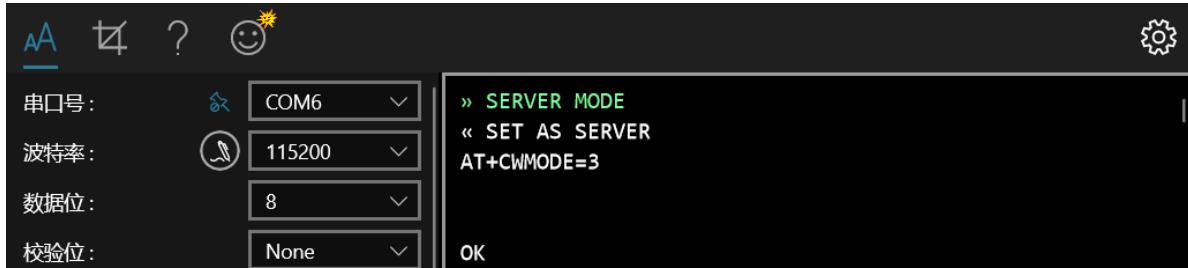
Parameter Settings

DMA Request	Channel	Direction	Priority
USART2_RX	DMA1 Channel 6	Peripheral To Memory	Low

## Configure the ESP8266 working mode

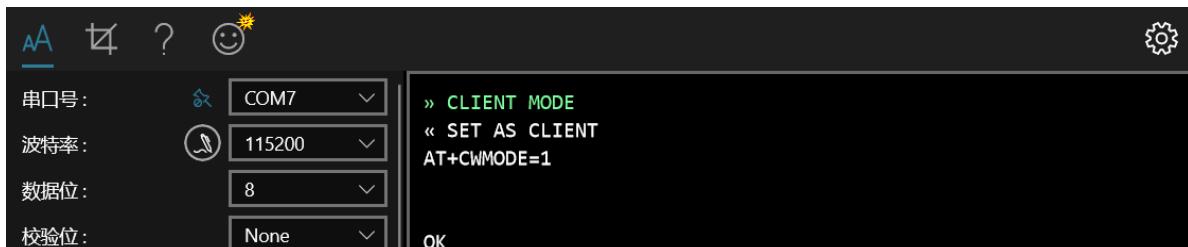
### SET as SERVER

- INPUT **SERVER MODE** in the serial port assistant helper to set the Wi-Fi as the **SERVER** mode.

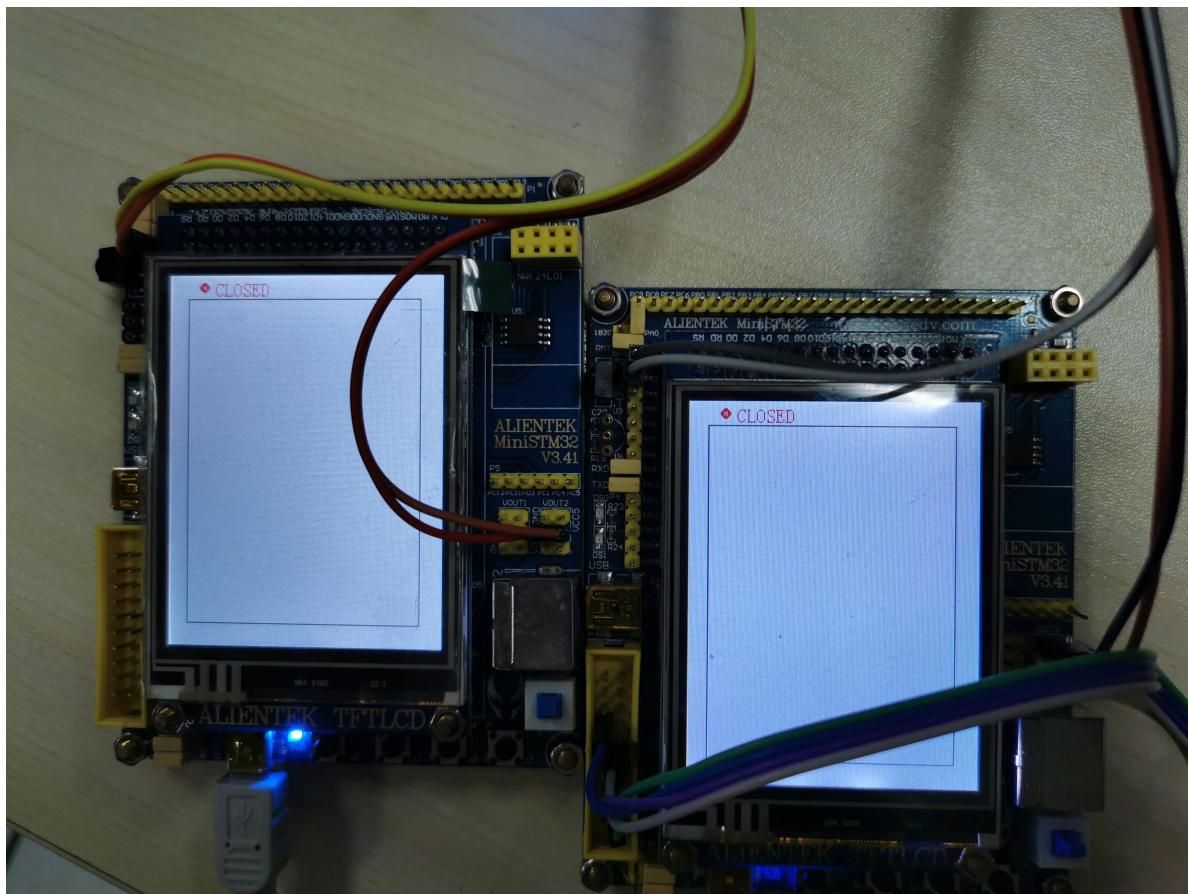


### SET as CLIENT

- INPUT **CLIENT MODE** in the serial port assistant helper to set the Wi-Fi as the **CLIENT** mode.



### Success example

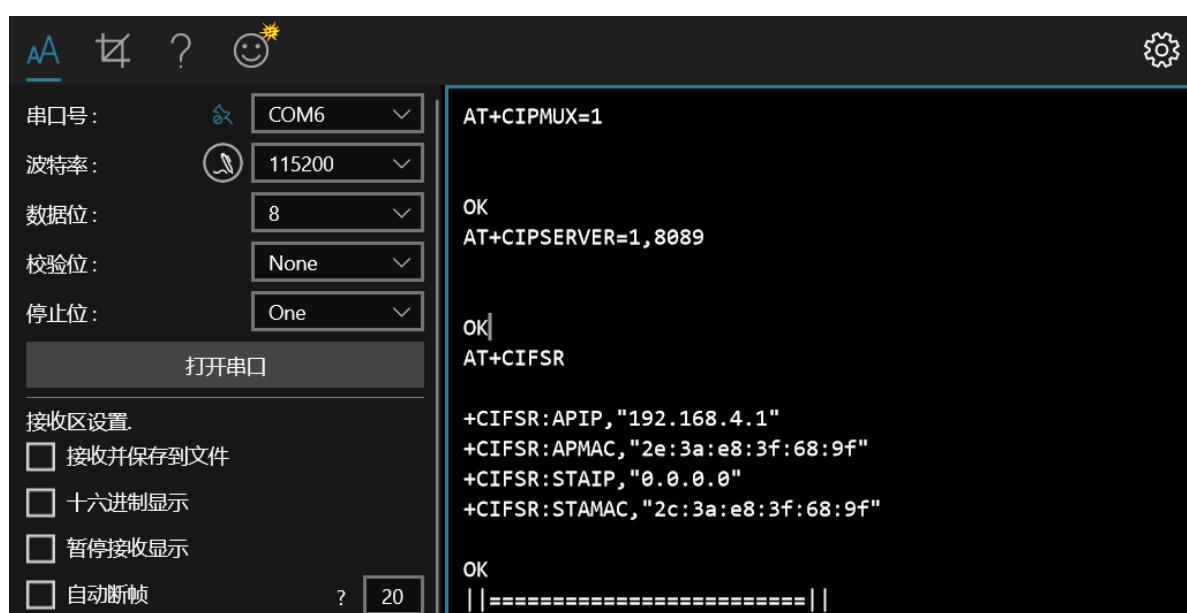
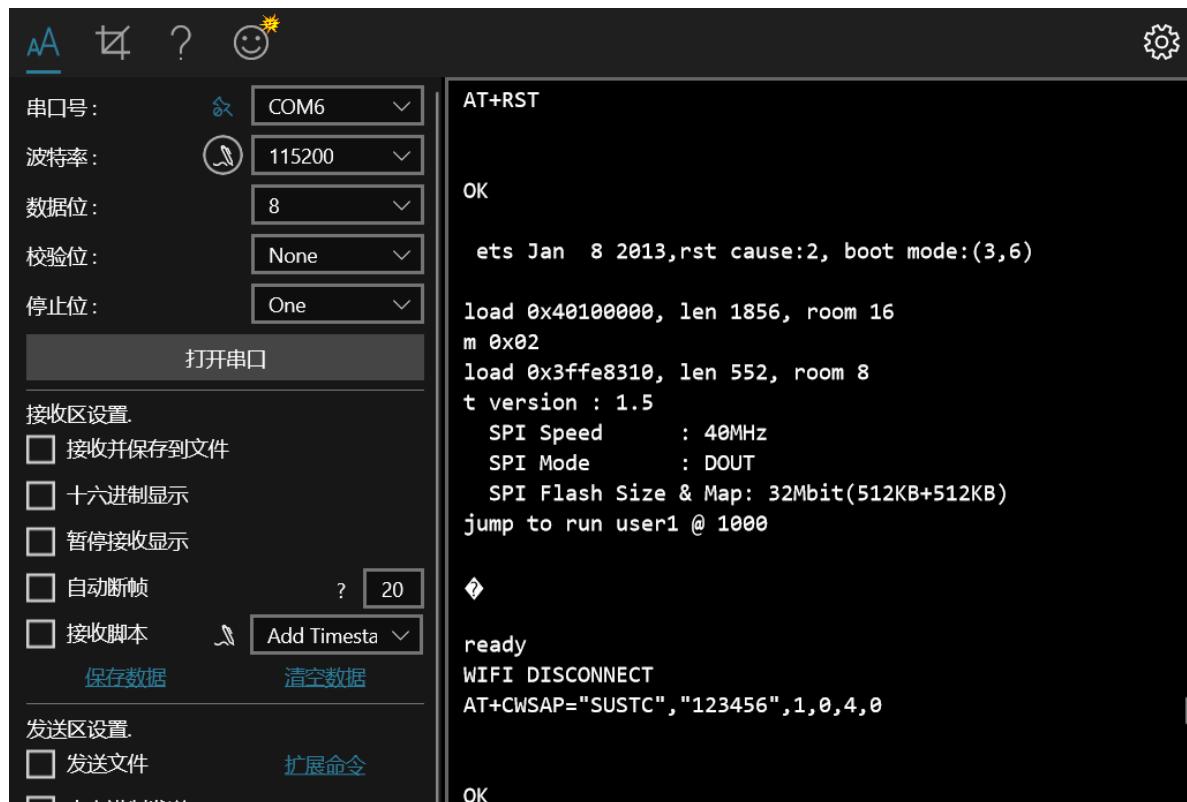


## Establish default connection

After configuring the ESP8266 working mode, we can press the **KEY0** to start the MiniSTM32 board and Wi-Fi module.

### First start the SERVER module

Press **KEY0** on the SERVER Wi-Fi to start it.



The Wi-Fi ssid is **SUSTC**, password is **123456**

### Second start the CLIENT module

Press **KEY0** on the CLIENT Wi-Fi to start it.

串口号: COM7  
波特率: 115200  
数据位: 8  
校验位: None  
停止位: One

接收区设置:  
 接收并保存到文件  
 十六进制显示  
 暂停接收显示  
 自动断帧 ? 20  
 接收脚本 Add Timesta

打开串口

```
AT+RST
OK
ets Jan 8 2013,rst cause:2, boot mode:(3,7)

load 0x40100000, len 1856, room 16
m 0x02
load 0x3ffe8310, len 552, room 8
t version : 1.5
    SPI Speed      : 40MHz
    SPI Mode       : DOUT
    SPI Flash Size & Map: 32Mbit(512KB+512KB)
jump to run user1 @ 1000

diamond

ready
```

串口号: COM7  
波特率: 115200  
数据位: 8  
校验位: None  
停止位: One

接收区设置:  
 接收并保存到文件  
 十六进制显示  
 暂停接收显示  
 自动断帧 ? 20  
 接收脚本 Add Timesta

打开串口

保存数据 清空数据

```
AT+CWJAP="SUSTC","123456"
OK
AT+CIPMUX=1

OK
AT+CIPSERVER=1,8089

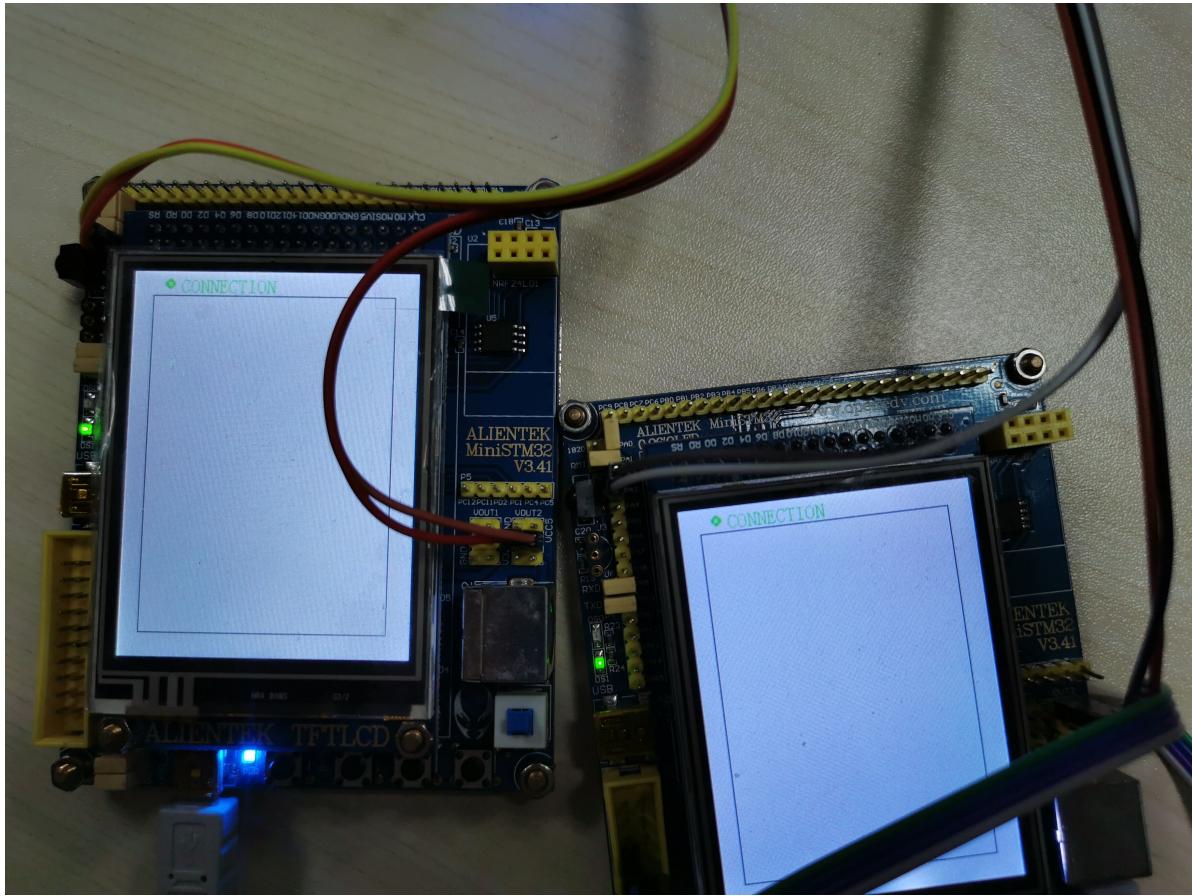
OK
AT+CIPSTART=0,"TCP","192.168.4.1",8089
Connection on id 0 ← Defined msg that shows the connection is on.
AT+CIFSR

+CIFSR:STAIP,"192.168.4.2"
+CIFSR:STAMAC,"2c:3a:e8:3f:b6:f3"

OK
```

## Success example

CONNECTION is on the screens.



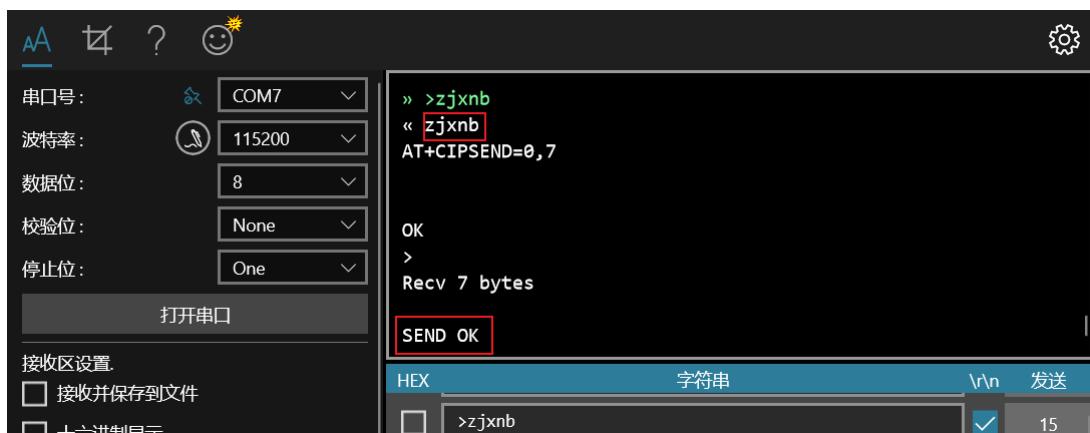
## Communicate with each other

Next, Happy communication!

If you want to communicate with each other, you must type the message in the textbox in the serial port assistant helper, and **most importantly**, when you want to send the message to the other, you must type the signal character **>** first. For example, if you want to send the message **zjxnb**, then you must type **>zjxnb** in the textbox, then you can see that **zjxnb** is sent and display in the both LCD screens.

### Success example

- Send **zjxnb** from the client Wi-Fi module to the server Wi-Fi module.



- Receive **zjxnb** from the server Wi-Fi module.

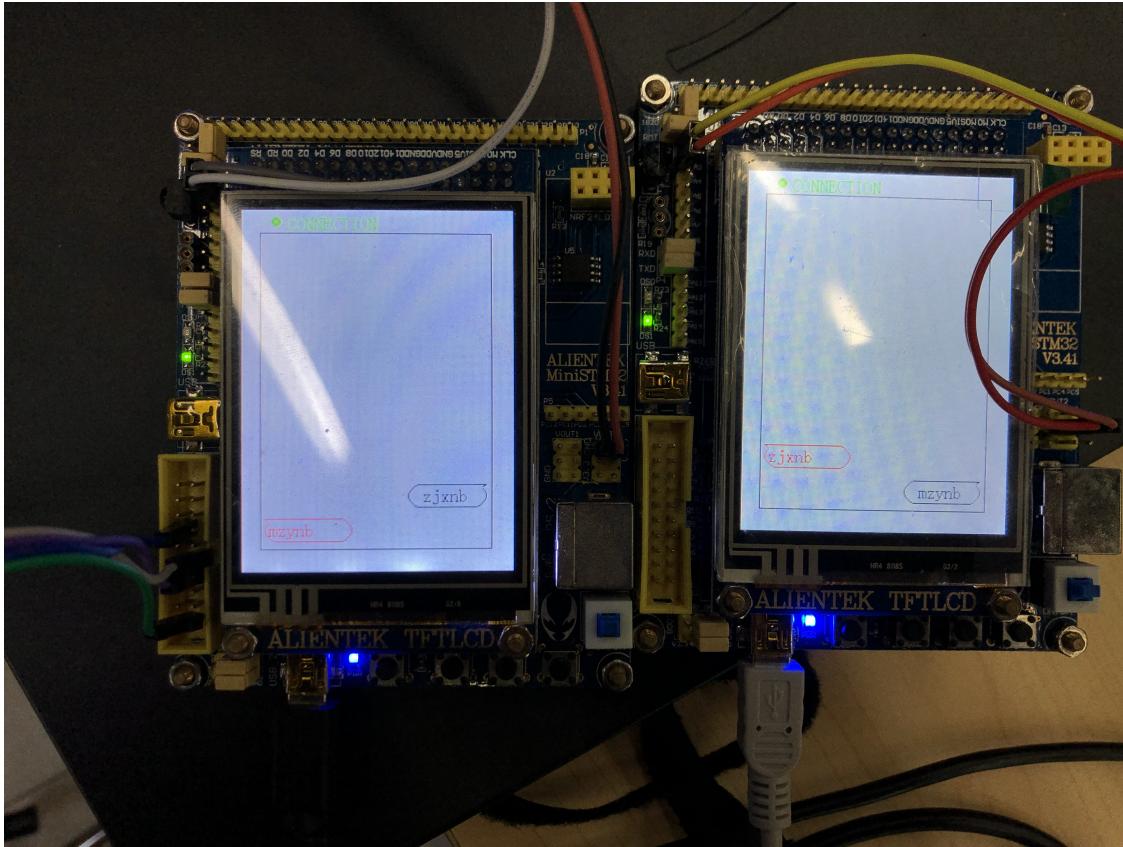


- Send `mzynb` from the server Wi-Fi module to the client Wi-Fi module.



- Example shown on the LCD screens of server Wi-Fi module and client Wi-Fi module.

The message sent to the other is shown on the **right** side, and the receiving message is shown on the **left** side.



## Disconnection

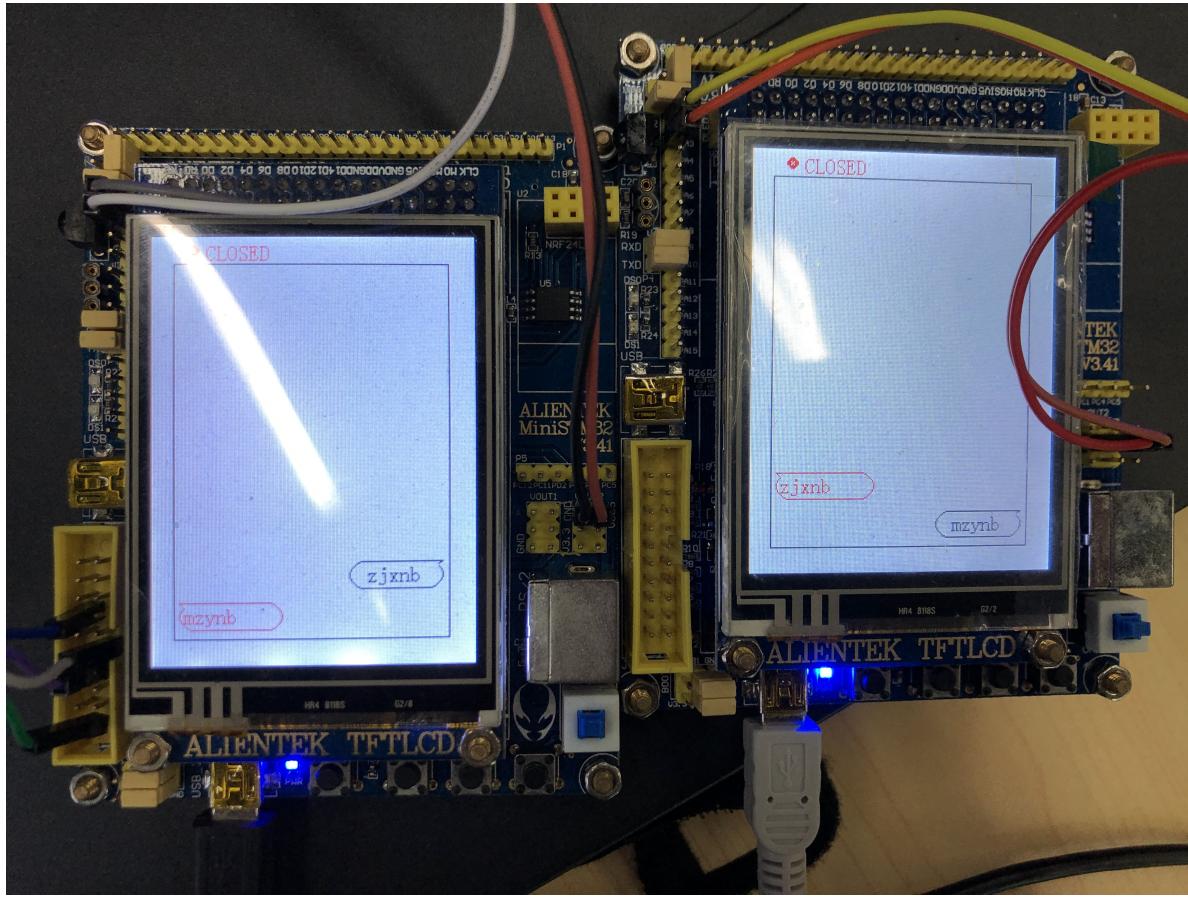
There are many ways to disconnect

- Auto disconnection.
- Plug out the Wi-Fi module.
- Type the command `AT+CIPCLOSE=0` to disconnect.
- Press `KEY0` to disconnect.
- Power off.

Except for powering off, the example picture is like this.

## Success Example

CLOSED is shown on the screens.



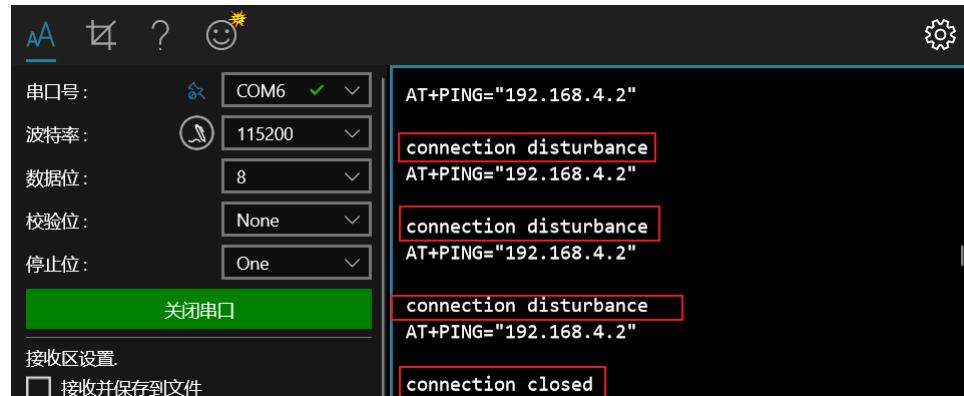
## Check connection

This part is for checking connection status due to long distances, inferences or shutting down.

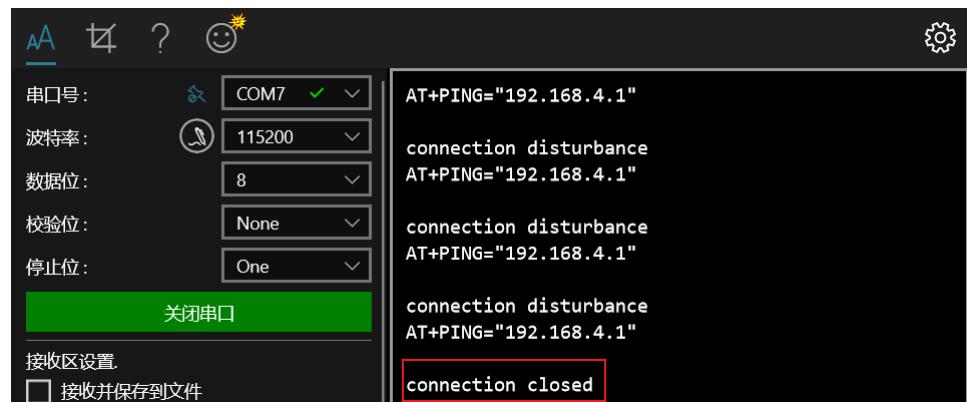
When the connection status is affected, then the `check connection function` takes effects. Using the command `AT+PING="X"` to check the connection between the two module. If failing for **four** times, it disconnects with the other device, and the status becomes **CLOSED**.

## Success example

- For the server mode



- For the client mode



## LED display

LED0	LED1	STATUS
OFF	OFF	Disconnection
OFF	ON	Wi-Fi connection but no TCP connection
ON	OFF	TCP connection
ON	ON	Sending/receiving message