6.2 WiFi module networking

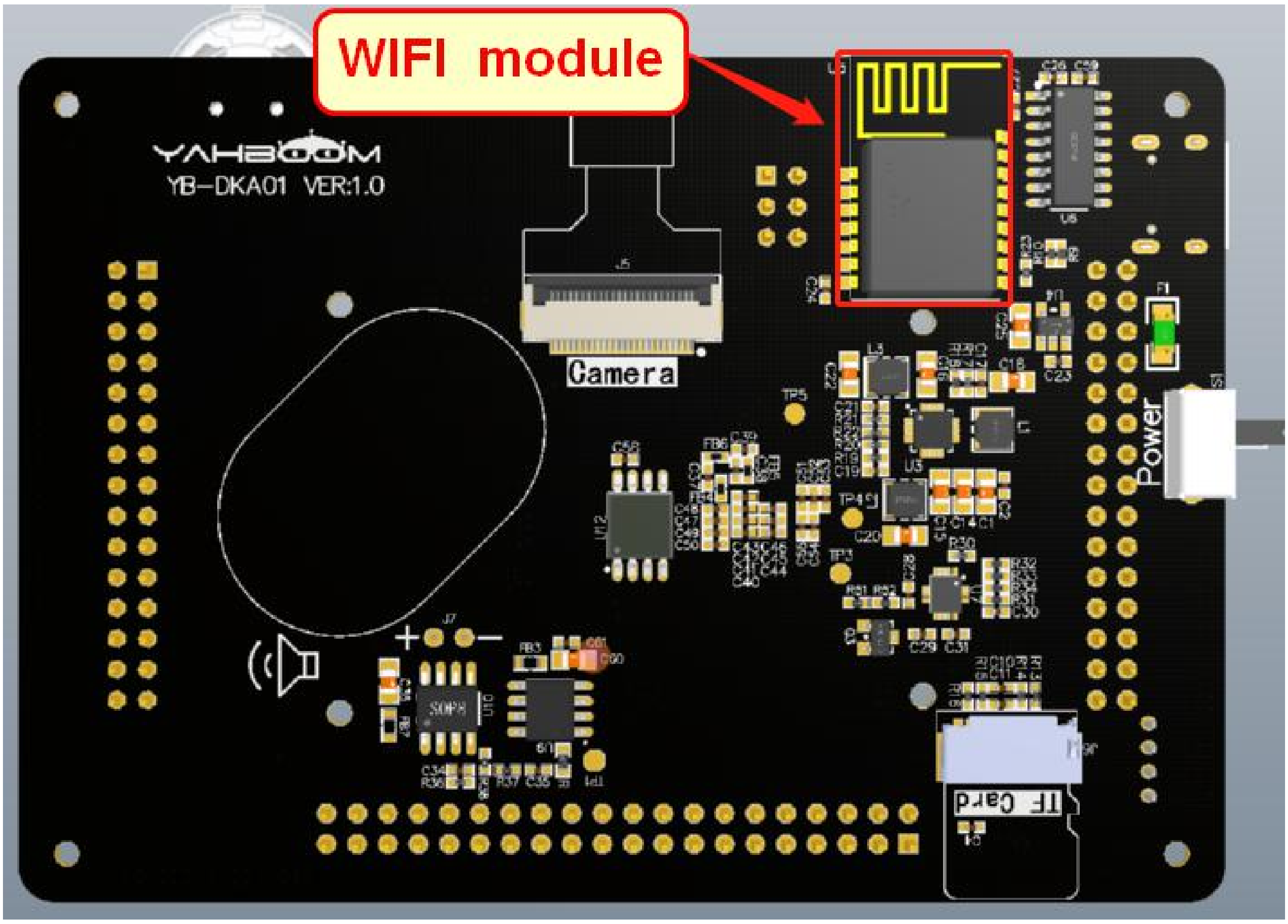
1. Experiment purpose

In this lesson, we mainly learn how to make WiFi module to connect to the network.

2.Experiment preparation

2.1 components

WiFi module



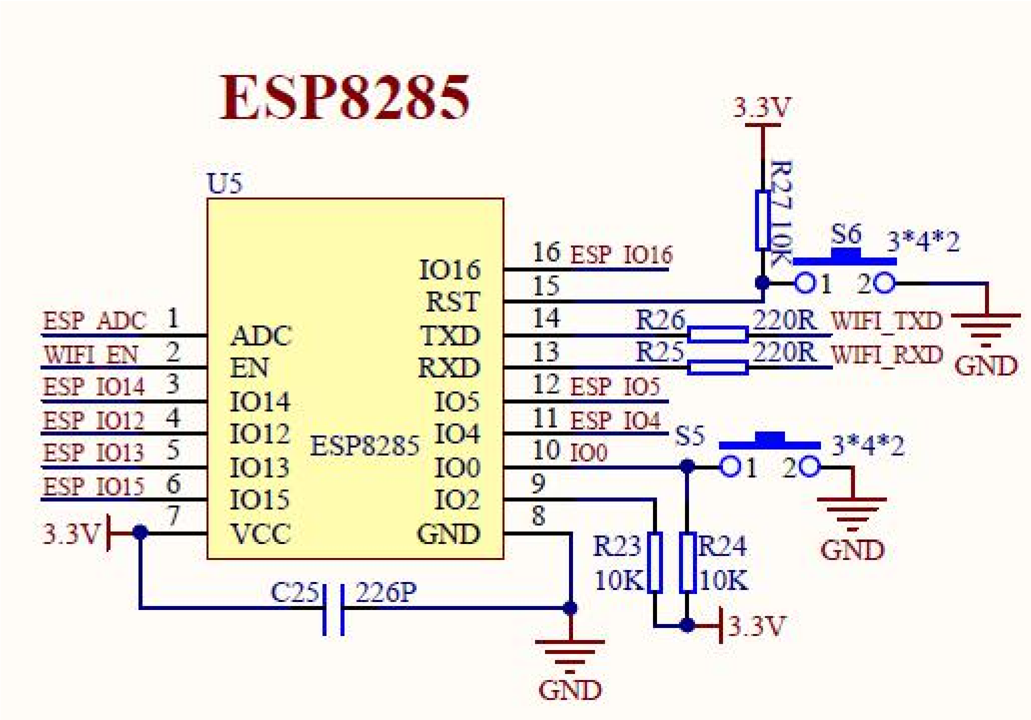
2.2 Component characteristics

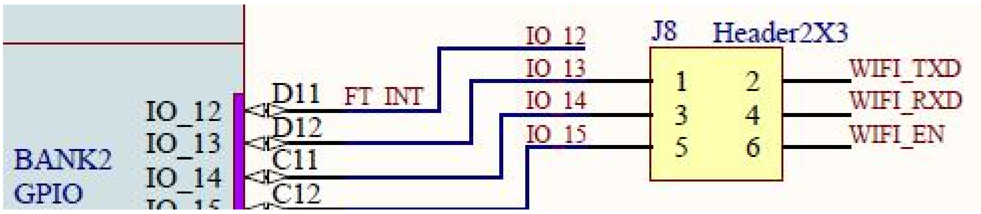
The WiFi module that comes with the K210 development board is the ESP8285 chip module.

2.3 Hardware connection

K210 development board has already been welded with the WiFi module by default. There are three jumper caps between the WiFi module and K210. If the jumper cap is inserted, the WIFI_TXD of ESP8285 is connected to IO13 of K210, WIFI_RXD is connected to IO14, WIFI_EN is connected to IO15, and the jumper is removed.

When we remove jumper caps, WIFI module will not work.





3. Experimental principle

WiFi module integrates an ESP8285 chip, which can be connected to the searched WiFi signal through AT commands.

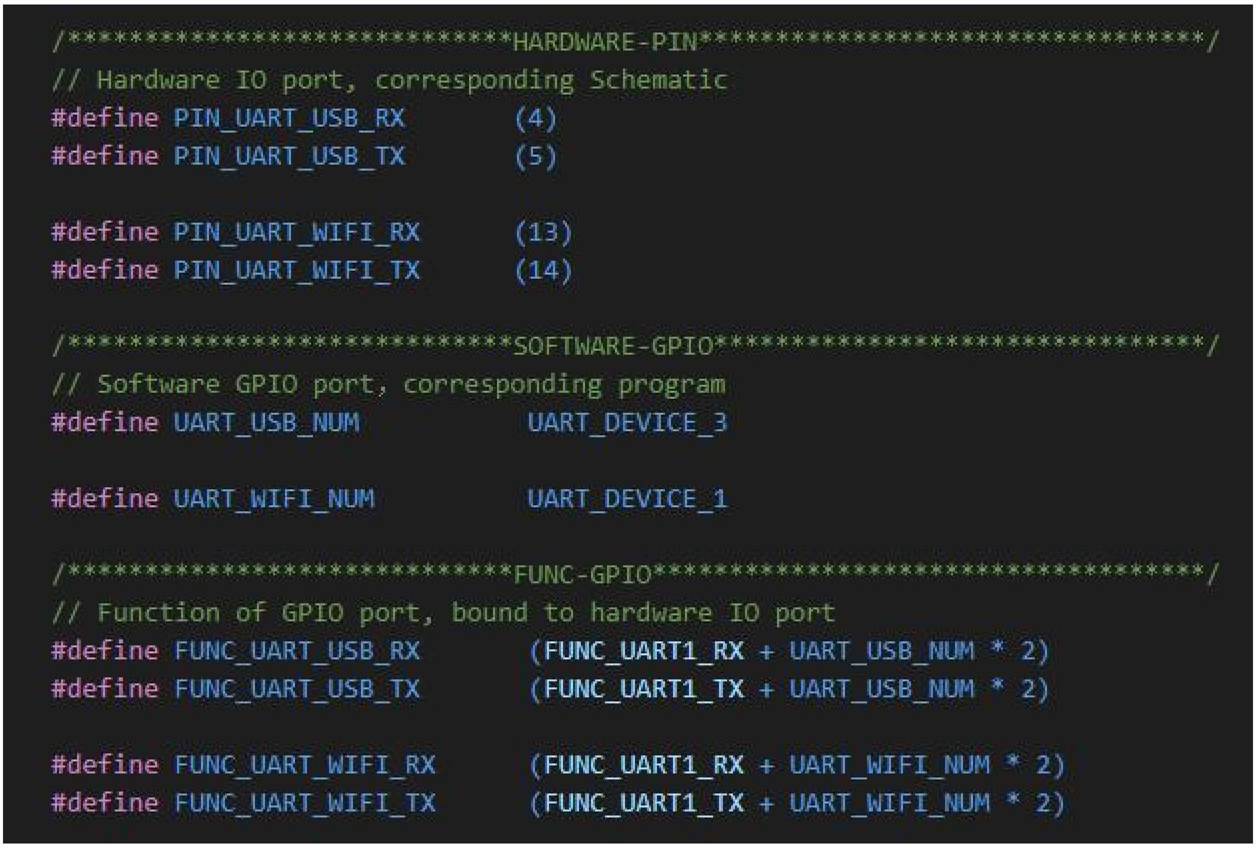
Through the function of data transmission through the serial port of K210, the data received by the serial assistant is transmitted to the WiFi module. The WiFi module can find the nearby WiFi signal and matches the password according to the content of the AT command, so that it can be connected to the WiFi router.

4. Experiment procedure

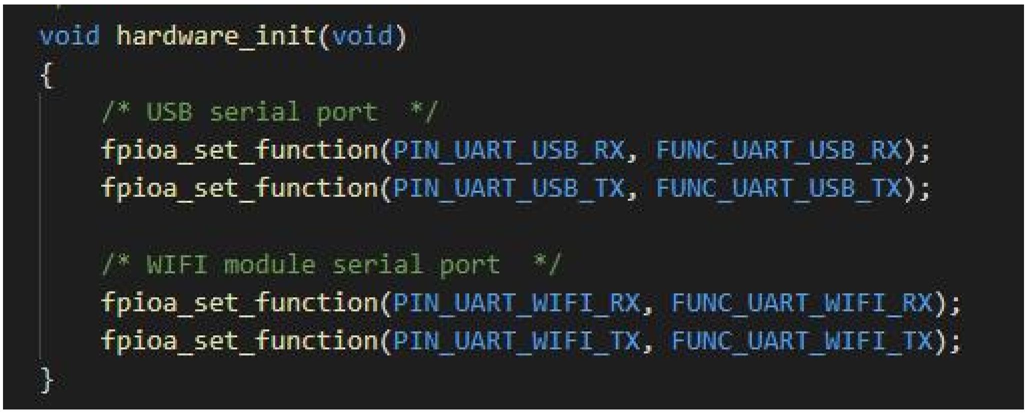
4.1 According to the above hardware connection pin diagram, K210 hardware pins and software functions use FPIOA mapping relationship.

IO4 and IO5 are the USB serial ports of the K210 development board, so they must be initialized.

The USB serial port uses serial port 3, and the serial port of the WiFi module uses serial port 1.

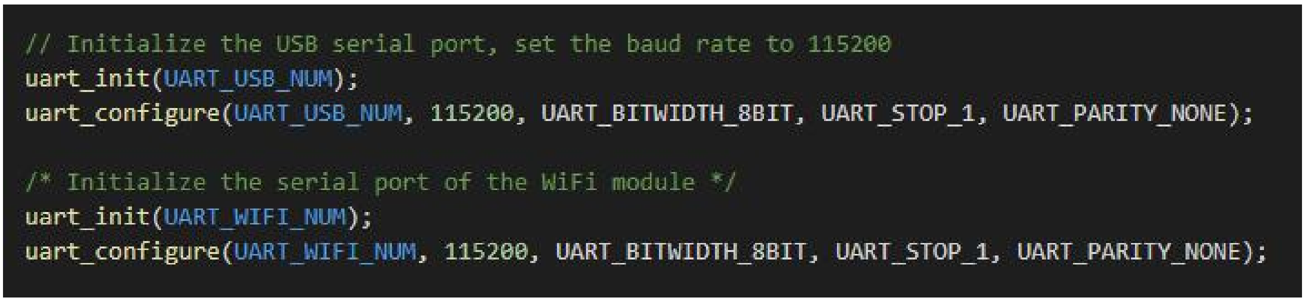


```
/*****************************HARDWARE-PIN*********************************/
// Hardware IO port, corresponding Schematic
#define PIN_UART_USB_RX       (4)
#define PIN_UART_USB_TX       (5)

#define PIN_UART_WIFI_RX      (13)
#define PIN_UART_WIFI_TX      (14)

/*****************************SOFTWARE-GPIO********************************/
// Software GPIO port, corresponding program
#define UART_USB_NUM           UART_DEVICE_3

#define UART_WIFI_NUM          UART_DEVICE_1

/*****************************FUNC-GPIO************************************/
// Function of GPIO port, bound to hardware IO port
#define FUNC_UART_USB_RX       (FUNC_UART1_RX + UART_USB_NUM * 2)
#define FUNC_UART_USB_TX       (FUNC_UART1_TX + UART_USB_NUM * 2)

#define FUNC_UART_WIFI_RX      (FUNC_UART1_RX + UART_WIFI_NUM * 2)
#define FUNC_UART_WIFI_TX      (FUNC_UART1_TX + UART_WIFI_NUM * 2)
```



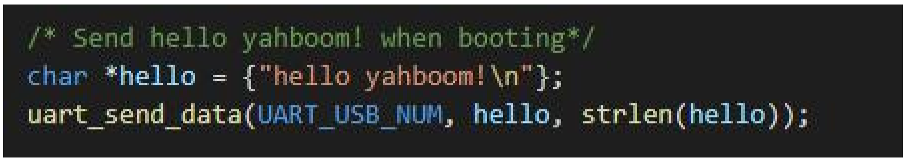
```
void hardware_init(void)
{
    /* USB serial port  */
    fpioa_set_function(PIN_UART_USB_RX, FUNC_UART_USB_RX);
    fpioa_set_function(PIN_UART_USB_TX, FUNC_UART_USB_TX);

    /* WIFI module serial port  */
    fpioa_set_function(PIN_UART_WIFI_RX, FUNC_UART_WIFI_RX);
    fpioa_set_function(PIN_UART_WIFI_TX, FUNC_UART_WIFI_TX);
}
```

4.2 Initialize the configuration of the serial port, baud rate is set to 115200, the serial port data width is 8 bits, stop bit is 1 bit, no parity.



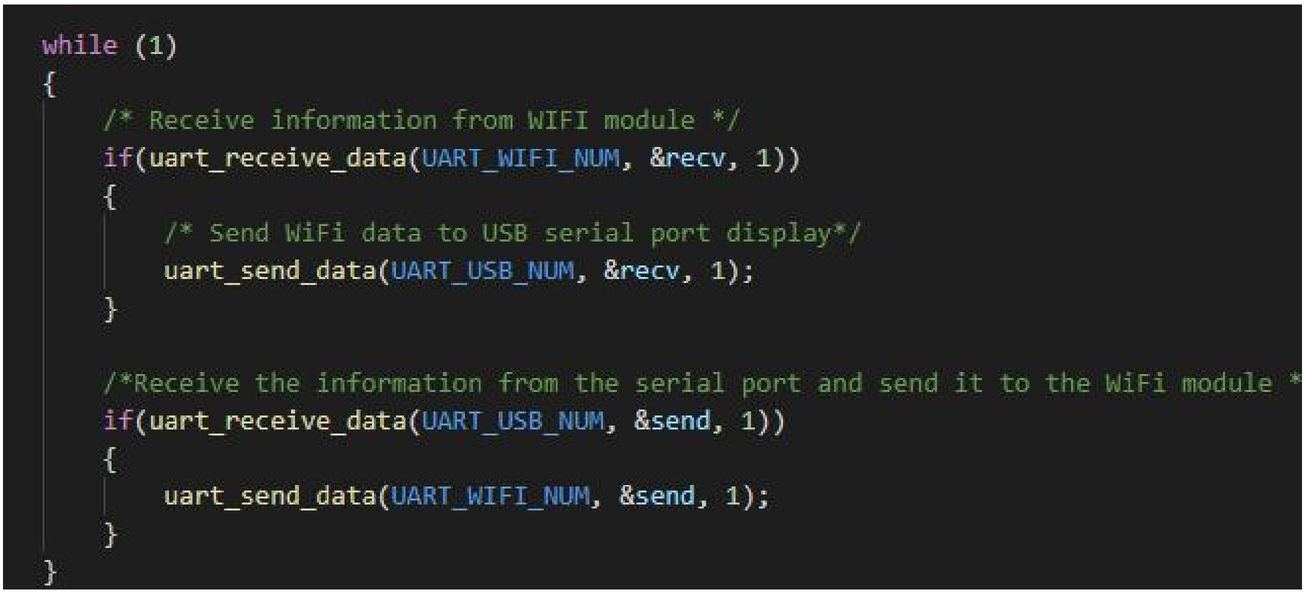
```
// Initialize the USB serial port, set the baud rate to 115200
uart_init(UART_USB_NUM);
uart_configure(UART_USB_NUM, 115200, UART_BITWIDTH_8BIT, UART_STOP_1, UART_PARITY_NONE);

/* Initialize the serial port of the WiFi module */
uart_init(UART_WIFI_NUM);
uart_configure(UART_WIFI_NUM, 115200, UART_BITWIDTH_8BIT, UART_STOP_1, UART_PARITY_NONE);
```

4.3 Send “hello yahboom!” when booting.



```
/* Send hello yahboom! when booting*/
char *hello = {"hello yahboom!\n"};
uart_send_data(UART_USB_NUM, hello, strlen(hello));
```

4.4 Finally, the data of the serial port is received cyclically. If the WiFi module has data transmitted to the K210 chip, the K210 will transmit the data to the computer through the serial port for display; if the serial port assistant on the computer sends data to the K210 chip, the K210 will also pass the data through the WiFi module The serial port is sent to the WiFi module.



```
while (1)
{
    /* Receive information from WIFI module */
    if(uart_receive_data(UART_WIFI_NUM, &recv, 1))
    {
        /* Send WiFi data to USB serial port display*/
        uart_send_data(UART_USB_NUM, &recv, 1);
    }

    /*Receive the information from the serial port and send it to the WiFi module *
    if(uart_receive_data(UART_USB_NUM, &send, 1))
    {
        uart_send_data(UART_WIFI_NUM, &send, 1);
    }
}
```

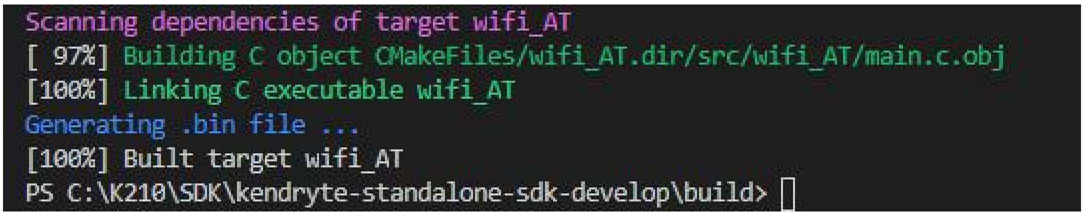
4.5 Compile and debug, burn and run

Copy the wifi_AT to the src directory in the SDK.

Then, enter the build directory and run the following command to compile.

cmake .. -DPROJ=wifi_AT -G "MinGW Makefiles"

make



```
Scanning dependencies of target wifi_AT
[ 97%] Building C object CMakeFiles/wifi_AT.dir/src/wifi_AT/main.c.obj
[100%] Linking C executable wifi_AT
Generating .bin file ...
[100%] Built target wifi_AT
PS C:\K210\SDK\kendryte-standalone-sdk-develop\build> 
```

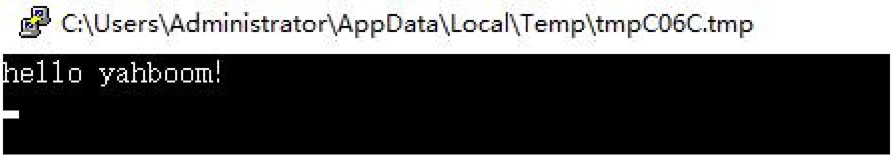
After the compilation is complete, the **wifi_AT.bin** file will be generated in the build folder.

We need to use the type-C data cable to connect the computer and the K210 development board.

Open kflash, select the corresponding device, and then burn the **wifi_AT.bin** file to the K210 development board.

5. Experimental phenomenon

5.1 After the firmware is write, a terminal interface will pop up. The gyroscope's data will be printed out. As shown below.



C:\Users\Administrator\AppData\Local\Temp\tmpC06C.tmp

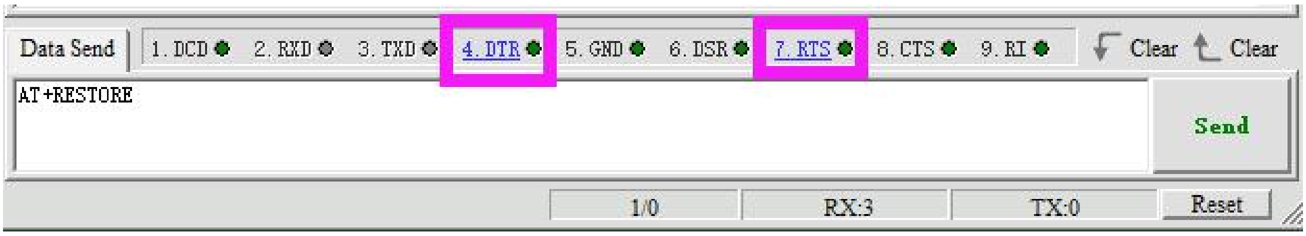
```
hello yahboom!
```

If you can't see terminal interface, please open the serial port assistant of the computer, select the corresponding serial port number of the corresponding K210 development board, set the baud rate to 115200.

5.2 Then, click to open the serial port assistant.

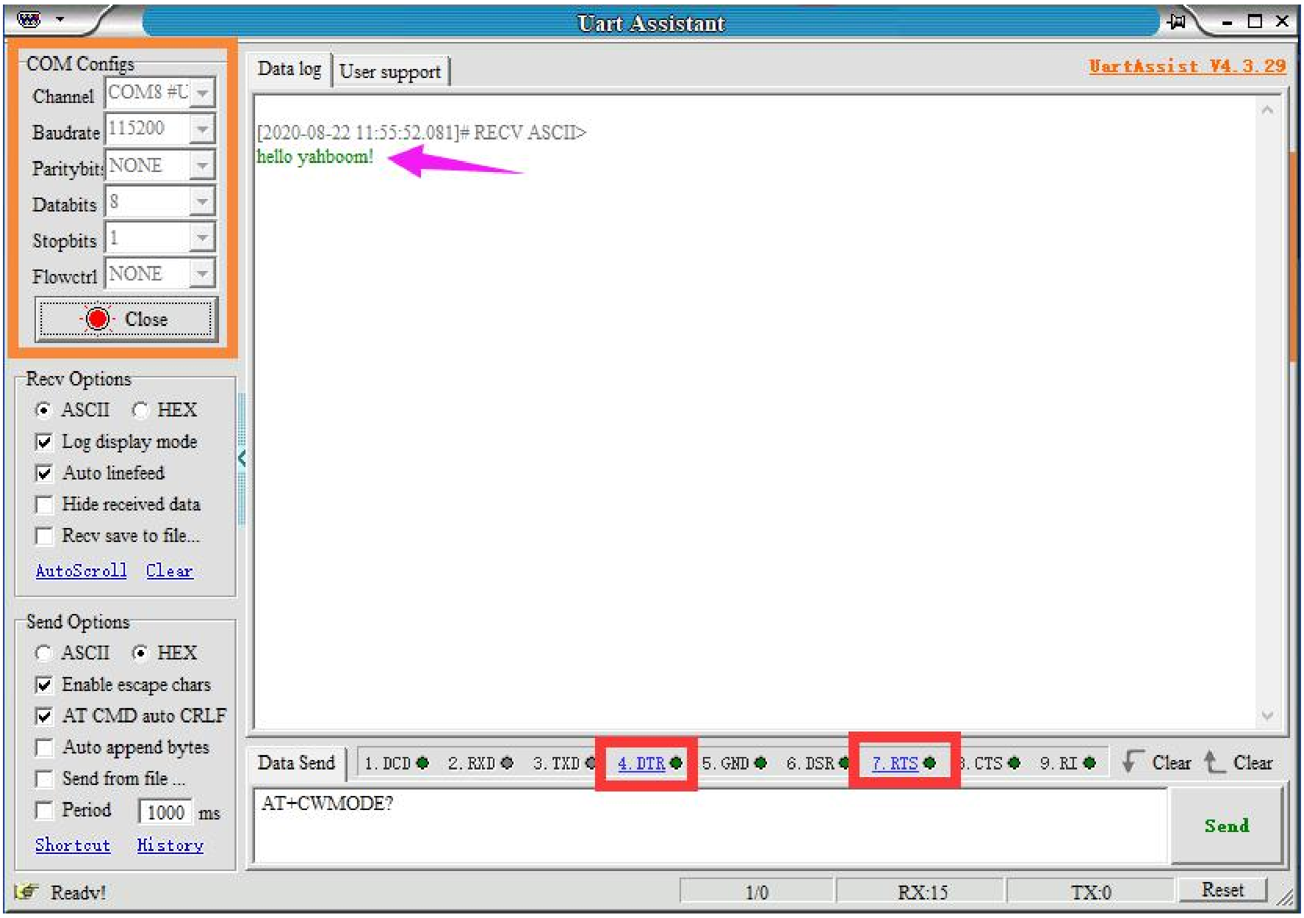
!Note: you also need to set the DTR and RTS of the serial port assistant.

Click 4.DTR and 7.RTS to set them to green.



```
AT+RESTORE
```

5.3 Press the **Reset button** on the K210 development board, and the serial debugging assistant will print "hello yahboom!".



COM Configs

Channel: COM8 #U

Baudrate: 115200

Paritybits: NONE

Databits: 8

Stopbits: 1

Flowctrl: NONE

Recv Options

● ASCII ○ HEX

☑ Log display mode

☑ Auto linefeed

☐ Hide received data

☐ Recv save to file...

Send Options

○ ASCII ● HEX

☑ Enable escape chars

☑ AT CMD auto CRLF

☐ Auto append bytes

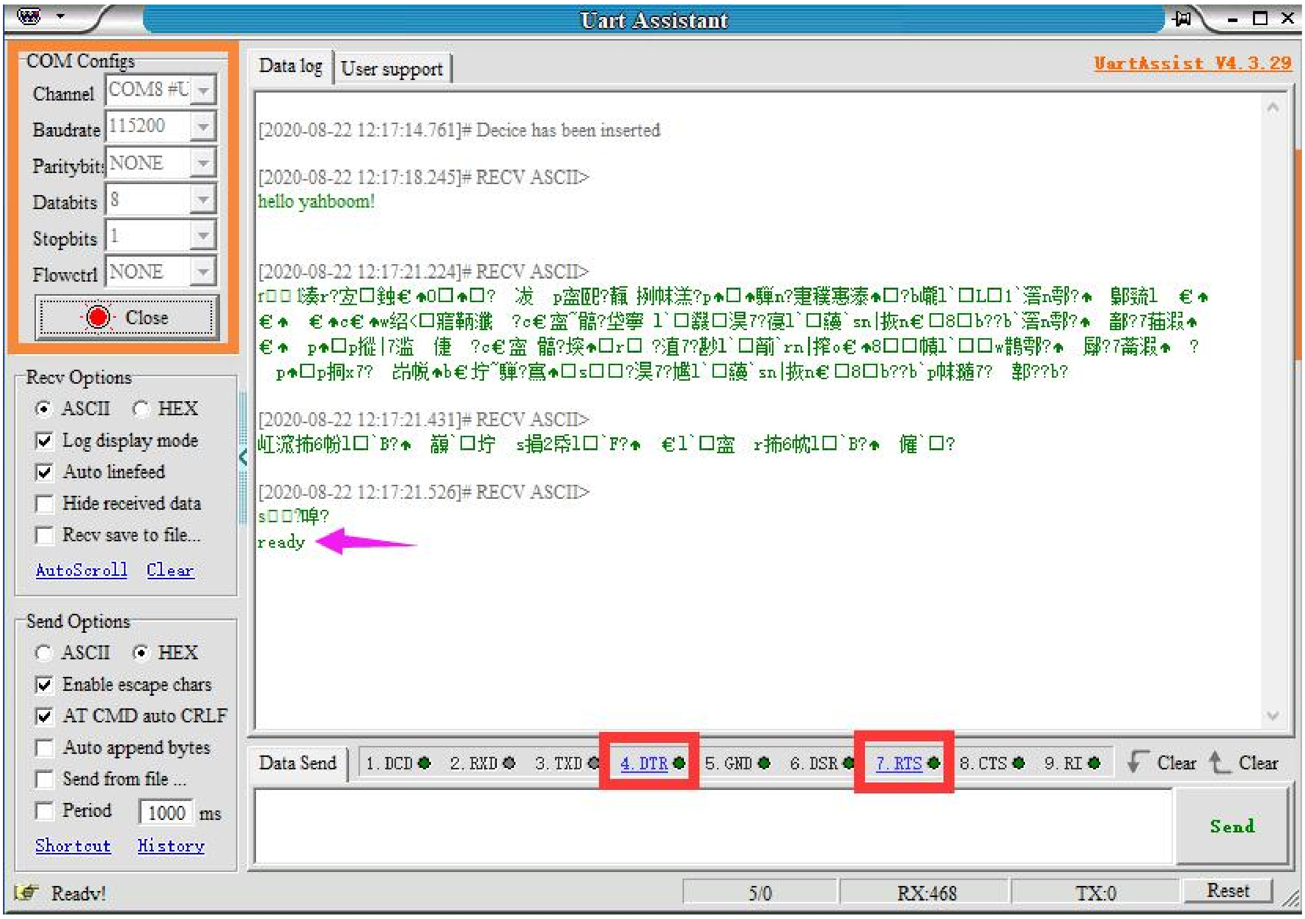
☐ Send from file ...

☐ Period 1000 ms

```
[2020-08-22 11:55:52.081]# RECV ASCII>
hello yahboom!
```

```
AT+CWMODE?
```

5.4 Then, press the **ESP-RST** button of the WiFi module, we will see the ready character, indicating that the WiFi module is normal.



Uart Assistant

UartAssist V4.3.29

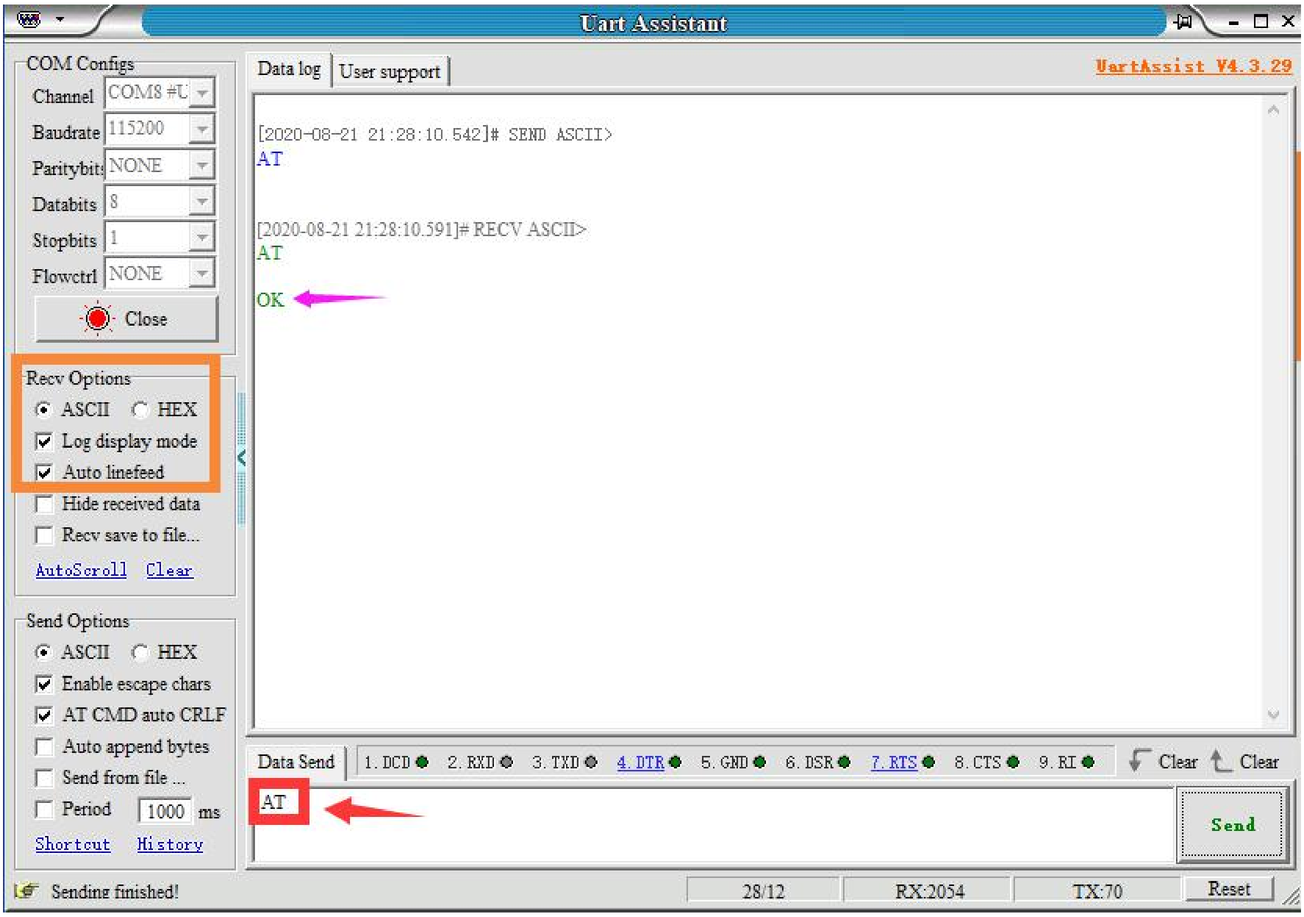
COM Configs
Channel: COM8 #U
Baudrate: 115200
Paritybits: NONE
Databits: 8
Stopbits: 1
Flowctrl: NONE

Recv Options
- ● ASCII ○ HEX
- ☑ Log display mode
- ☑ Auto linefeed
- ☐ Hide received data
- ☐ Recv save to file...

Send Options
- ○ ASCII ● HEX
- ☑ Enable escape chars
- ☑ AT CMD auto CRLF
- ☐ Auto append bytes
- ☐ Send from file ...
- ☐ Period 1000 ms

```
[2020-08-22 12:17:14.761]# Decice has been inserted

[2020-08-22 12:17:18.245]# RECV ASCII>
hello yahboom!


[2020-08-22 12:17:21.224]# RECV ASCII>
r□□ 獉r?宜□鈍€♠0□♠□?  泼  p盗郎?藾 捌帏渼?p♠□♠驆n?疐樸惠溙♠□?b巃l`□L□1`滘n鄠?♠  鄓巰l  €♠
€♠  €♠c€♠w绍<□寤鞆瀐  ?c€盗˜髗?垒寧 l`□韄□淏7?復l`□蘟`sn|扳n€□8□b??b`滘n鄠?♠  鄱?7葹溵♠
€♠  p♠□p摐|7滥  倢  ?c€盗 髗?墺♠□r□ ?湆7?尠l`□蒯`rn|搾o€♠8□□幀l`□□w鶻鄠?♠  鄏?7蒍溵♠  ?
  p♠□p摑x7?  岀帨♠b€拧˜驆?寘♠□s□□?淏7?摅l`□蘟`sn|扳n€□8□b??b`p帏黐7?  鄣??b?

[2020-08-22 12:17:21.431]# RECV ASCII>
屸潹捬6帉1□`B?♠  巑`□拧  s揖2帍1□`F?♠  €l`□盗  r捬6帺1□`B?♠  儱`□?

[2020-08-22 12:17:21.526]# RECV ASCII>
s□□?啤?
ready
```

Enter AT characters in the send box, then, click “send”, the serial port assistant will display “OK”.



COM Configs
Channel: COM8 #U
Baudrate: 115200
Paritybits: NONE
Databits: 8
Stopbits: 1
Flowctrl: NONE

Recv Options
● ASCII ○ HEX
☑ Log display mode
☑ Auto linefeed
□ Hide received data
□ Recv save to file...

Send Options
● ASCII ○ HEX
☑ Enable escape chars
☑ AT CMD auto CRLF
□ Auto append bytes
□ Send from file ...
□ Period 1000 ms

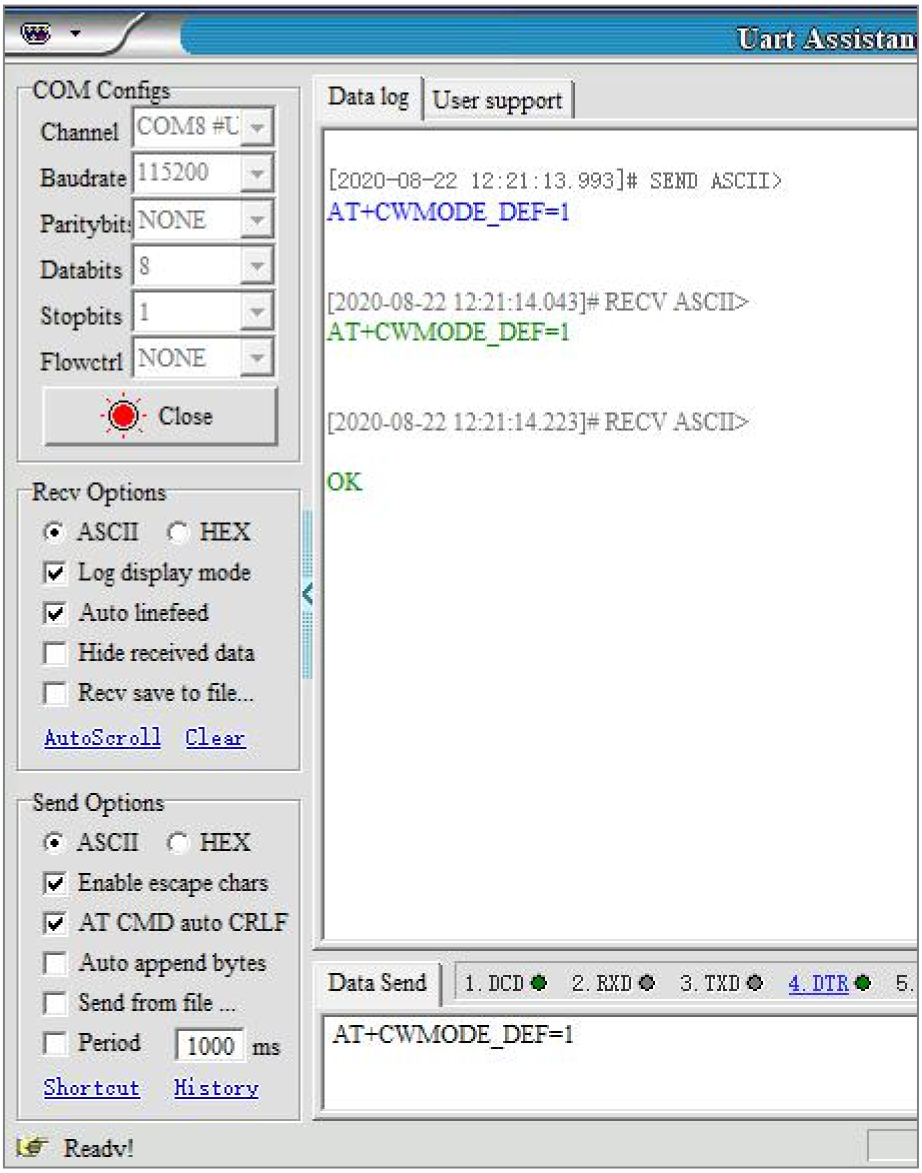
```
[2020-08-21 21:28:10.542]# SEND ASCII>
AT

[2020-08-21 21:28:10.591]# RECV ASCII>
AT

OK
```

```
AT
```

Sending finished!   28/12   RX:2054   TX:70

5.5 Send following AT commands to set the WiFi module mode to workstation.

AT+CWMODE_DEF=1



Channel: COM8 #U
Baudrate: 115200
Paritybits: NONE
Databits: 8
Stopbits: 1
Flowctrl: NONE

Recv Options: ● ASCII ○ HEX
☑ Log display mode
☑ Auto linefeed
□ Hide received data
□ Recv save to file...

Send Options: ● ASCII ○ HEX
☑ Enable escape chars
☑ AT CMD auto CRLF
□ Auto append bytes
□ Send from file ...
□ Period 1000 ms

```
[2020-08-22 12:21:13.993]# SEND ASCII>
AT+CWMODE_DEF=1


[2020-08-22 12:21:14.043]# RECV ASCII>
AT+CWMODE_DEF=1


[2020-08-22 12:21:14.223]# RECV ASCII>

OK
```

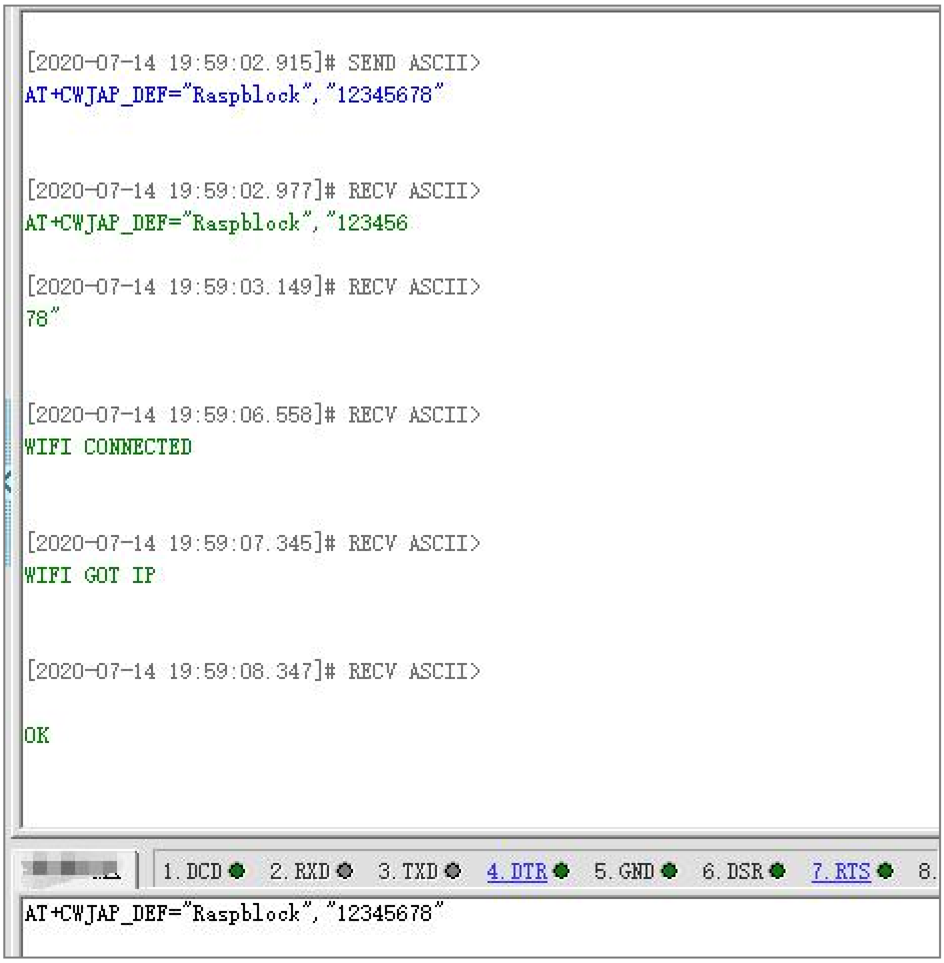
```
AT+CWMODE_DEF=1
```

5.6 Send following AT commands to set the WiFi signal need to connect.

AT+CWJAP_DEF="WiFi name","password"

Eg: AT+CWJAP_DEF="Raspblock","12345678"

If we receive “WIFI CONNECTED” and “WIFI GOT IP” , it means WIFI is successfully connected.

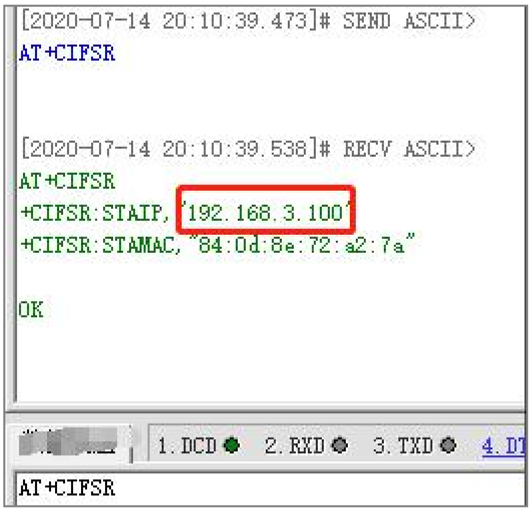


```
[2020-07-14 19:59:02.915]# SEND ASCII>
AT+CWJAP_DEF="Raspblock","12345678"


[2020-07-14 19:59:02.977]# RECV ASCII>
AT+CWJAP_DEF="Raspblock","123456

[2020-07-14 19:59:03.149]# RECV ASCII>
78"


[2020-07-14 19:59:06.558]# RECV ASCII>
WIFI CONNECTED


[2020-07-14 19:59:07.345]# RECV ASCII>
WIFI GOT IP


[2020-07-14 19:59:08.347]# RECV ASCII>

OK
```

```
AT+CWJAP_DEF="Raspblock","12345678"
```

5.7 Get the current WIFI module IP address

AT+CIFSR

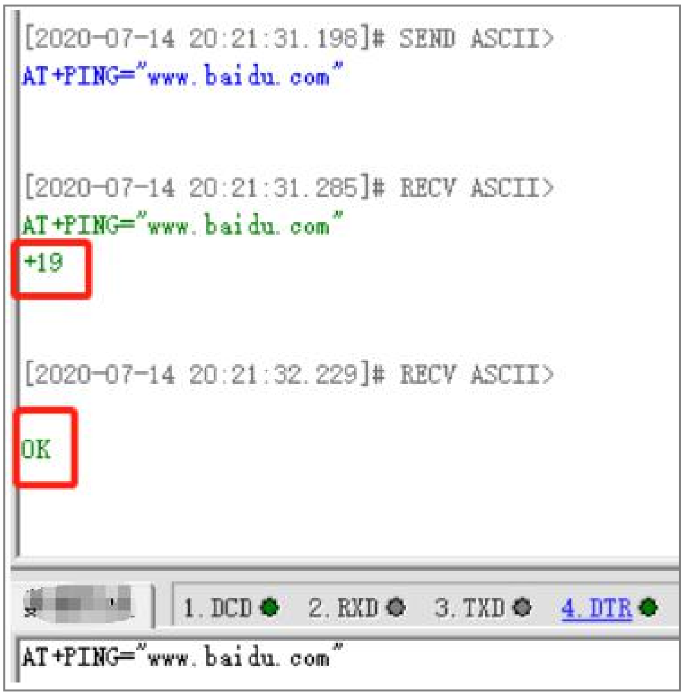


```
[2020-07-14 20:10:39.473]# SEND ASCII>
AT+CIFSR


[2020-07-14 20:10:39.538]# RECV ASCII>
AT+CIFSR
+CIFSR:STAIP,"192.168.3.100"
+CIFSR:STAMAC,"84:0d:8e:72:a2:7a"

OK
```

```
AT+CIFSR
```

5.8 **ping** the website and check whether the network can be connected. A value indicates that the network is normal.

AT+ping="xxx"



```
[2020-07-14 20:21:31.198]# SEND ASCII>
AT+PING="www.baidu.com"

[2020-07-14 20:21:31.285]# RECV ASCII>
AT+PING="www.baidu.com"
+19

[2020-07-14 20:21:32.229]# RECV ASCII>

OK
```

```
AT+PING="www.baidu.com"
```

6. Experiment summary

6.1 The WiFi module communicates with K210 through a serial port connection.

6.2 The K210 only be used to transfer data during this process, it will not process data.

6.3 If the WiFi module is connected to other serial chips, it can also be operated by AT commands.

6.4 After the K210 development board is successfully connected to WIFI, it will automatically connect to this WiFi signal after the next restart.