

AT Command Examples of WT32-ETH01 Wired Module

Version 1.1

Contents

Wired Network.....	2
1.TCP Client Connection.....	2
2.TCP Server Connection.....	4
3.UDP Client Connection.....	6
4.UDP Server Connection.....	8
WiFi Connection.....	10
1.TCP Client Connection.....	10
2.TCP Server Connection.....	12
3.UDP Client Connection.....	14
4.UDP Server Connection.....	16
HTTP Request.....	18
Serial to Bluetooth Transparent Data Transmission.....	20
Bluetooth to WiFi Transparent Data Transmission.....	22
WiFi to Ethernet Transparent Data Transmission.....	25

Wired Network

1. TCP Client Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=1 //Serial to Ethernet mode (default mode)
```

Response:

OK

2. Set up DHCP

```
AT+CWDHCP_DEF=3,0
```

Response:

OK

3. Set IP address

```
AT+CIPETH_DEF="192.168.0.7","192.168.0.1","255.255.255.0"
```

Response:

OK

Note:

When setting to static IP , please write IP , gateway and subnet mask at the same time (the default static IP: 192.168.0.7).

When setting to DHCP/ Dynamic IP, it will automatically obtain IP and other related information.

When the module is directly connected to the computer, it cannot be set to DHCP/ Dynamic IP . Generally, the computer does not have the ability to assign IP address. If the module is set to DHCP directly connected to the computer, the module will be in a state of waiting for the IP address to be assigned, which will cause the module to fail to perform normal transparent transmission.

4. Query the device's IP address

```
AT+CIPETH_DEF?
```

Response:

```
+CIPETH_DEF:"192.168.0.7","192.168.0.1","255.255.255.0"
```

OK

5. Connect the module directly to the computer via Ethernet to establish a TCP server.

For example: IP:192.168.0.201; port: 8080

6. The module is connected to the server as a TCP client.

```
AT+CIPSTART="TCPC","192.168.0.201",8080//protocol, server IP and port
```

Response:

OK

7. The module sends data to the server.

AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes

>test // enter the data, no CR

Response:

Recv 4 bytes

SEND OK

8. Receive the server data

Response:

+IPD,n:xxxxxxxxxx // received n bytes, data=xxxxxxxxxx

9. Enable the transparent transmission mode

AT+CIPMODE=1//

Response:

OK

10. Start sending data

AT+CIPSEND

>

11. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized, the transparent transmission will be exited.

12. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

13. Close the TCP connection

AT+CIPCLOSE

Response:

CLOSED

OK

2. TCP Server Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=1 //Serial to Ethernet mode (default mode)
```

Response:

OK

2. Set up DHCP

```
AT+CWDHCP_DEF=3,0
```

Response:

OK

3. Set the IP address

```
AT+CIPETH_DEF="192.168.0.7","192.168.0.1","255.255.255.0"
```

Response:

OK

Note:

When setting to static IP , please write IP , gateway and subnet mask at the same time (the default static IP: 192.168.0.7).

When setting to DHCP/ Dynamic IP, it will automatically obtain IP and other related information.

When the module is directly connected to a computer, it cannot be set to DHCP/dynamic IP. Generally, computers do not have the ability to assign IP addresses.

If the module is set to DHCP directly connected to the computer, it will cause the module to be in a state of waiting for the IP address to be assigned, which will cause the module to fail to perform normal transparent transmission.

4. Query the device's IP address

```
AT+CIPETH_DEF?
```

Response:

```
+CIPETH_DEF:"192.168.0.7","192.168.0.1","255.255.255.0"
```

OK

5. Create a TCP server

```
AT+CIPSTART="TCPS","192.168.0.201",8080,3333//local port 3333
```

6. The module sends data to the client

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:

Recv 4 bytes

SEND OK

7. Receive the client data

Response:

+IPD,n:xxxxxxxxxx // received n bytes, data=xxxxxxxxxx

8. Enable the transparent transmission mode

AT+CIPMODE=1//

Response:

OK

9. Start sending data

AT+CIPSEND

>

10. Exit sending data:

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

11. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

12. Close the connection

AT+CIPCLOSE

Response:

CLOSED

OK

3.UDP Client Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=1 //Serial to Ethernet mode (default mode)
```

Response:

OK

2. Set up DHCP

```
AT+CWDHCP_DEF=3,0
```

Response:

OK

3. Set the IP address

```
AT+CIPETH_DEF="192.168.0.7","192.168.0.1","255.255.255.0"
```

Response:

OK

Note:

When setting to static IP , please write IP , gateway and subnet mask at the same time (the default static IP: 192.168.0.7).

When setting to DHCP/ Dynamic IP, it will automatically obtain IP and other related information.

When the module is directly connected to a computer, it cannot be set to DHCP/dynamic IP. Generally, computers do not have the ability to assign IP addresses.

If the module is set to DHCP directly connected to the computer, it will cause the module to be in a state of waiting for the IP address to be assigned, which will cause the module to fail to perform normal transparent transmission.

4. Query the device's IP address

```
AT+CIPETH_DEF?
```

Response:

```
+CIPETH_DEF:"192.168.0.7","192.168.0.1","255.255.255.0"
```

OK

5. The module is directly connected to the PC via Ethernet to create a UDP server

For example: IP: 192.168.0.201; port: 8080

6. Connect the module to the server as a UDP client

```
AT+CIPSTART="UDPC","192.168.0.201",8080,3333
```

Response:

OK

7. The module sends data to the server

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:
Recv 4 bytes
SEND OK

8. Receive the server data

Response:
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx

9. Enable the transparent transmission mode

AT+CIPMODE=1//

Response:

OK

10. Start sending data

AT+CIPSEND

>

11. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

12. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

13. Close the TCP connection

AT+CIPCLOSE

Response:

CLOSED

OK

4.UDP Server Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=1 //Serial to Ethernet mode (default mode)
```

Response:

OK

2. Set up DHCP

```
AT+CWDHCP_DEF=3,0
```

Response:

OK

3. Set the IP address

```
AT+CIPETH_DEF="192.168.0.7","192.168.0.1","255.255.255.0"
```

Response:

OK

Note:

When setting to static IP , please write IP , gateway and subnet mask at the same time (the default static IP: 192.168.0.7).

When setting to DHCP/ Dynamic IP, it will automatically obtain IP and other related information.

When the module is directly connected to a computer, it cannot be set to DHCP/dynamic IP. Generally, computers do not have the ability to assign IP addresses. If the module is set to DHCP and directly connected to the computer, it will cause the module to be in a state of waiting for the IP address to be assigned, which will cause the module to fail to perform normal transparent transmission.

4. Query the device's IP address

```
AT+CIPETH_DEF?
```

Response:

```
+CIPETH_DEF:"192.168.0.7","192.168.0.1","255.255.255.0"
```

OK

5. The module is directly connected to the PC through Ethernet to establish UDP.

For example: IP: 192.168.0.201; port: 8080

6. The module establishes a UDP connection as a UDP server.

```
AT+CIPSTART="UDPS","192.168.0.201",8080,3333
```

Response:

OK

7. The module sends data to the client.

```
AT+CIPSEND=4 // set date length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:

Recv 4 bytes

SEND OK

8. Receive the server data

Response:

+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx

9. Enable the transparent transmission mode

AT+CIPMODE=1//

Response:

OK

10. Start sending data

AT+CIPSEND

>

11. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized, the transparent transmission will be exited.

12. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

13. Close the TCP connection

AT+CIPCLOSE

Response:

CLOSED

OK

WiFi Connection

1. TCP Client Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=2 //Serial to WiFi mode
```

Response:

OK

2. Connect to a router

```
AT+CWJAP="SSID","password"
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:STA,"192.168.1.103"
```

```
+CIFSR:STAMAC,"24:0a:c4:2a:25:8c"
```

```
+CIFSR:ETHIP,"0.0.0.0"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Connect the computer and the module to the same router, and use a network debugging tool on the PC to create a TCP server.

For example: IP:192.168.1.101; port:8888

5. The module is connected to the server as a TCP client.

```
AT+CIPSTART="TCPC","192.168.1.101",8888//protocol, server IP and port
```

Response:

OK

6. The module sends data to the server.

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:

```
Recv 4 bytes
```

```
SEND OK
```

7. Receive the server data

Response:

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

8. Enable the transparent transmission mode

AT+CIPMODE=1//

Response:

OK

9. Start sending data

AT+CIPSEND

>

10. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

11. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

12. Close the TCP connection

AT+CIPCLOSE

Response:

CLOSED

OK

2. TCP Server Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=2 //Serial to WiFi mode
```

Response:

OK

2. Connect to a router

```
AT+CWJAP="SSID","password"
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:STA,"192.168.1.103"
```

```
+CIFSR:STAMAC,"24:0a:c4:2a:25:8c"
```

```
+CIFSR:ETHIP,"0.0.0.0"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Create a TCP server

```
AT+CIPSTART="TCPS","192.168.0.201",8080,3333//Local port3333
```

5. Connect the computer and the module to the same router, and use a network debugging tool on PC to establish a TCP connection.

6. The module sends data to the client.

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:

```
Recv 4 bytes
```

```
SEND OK
```

7. Receive the client data

Response:

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

8. Enable the transparent transmission mode

```
AT+CIPMODE=1//
```

Response:

OK

9. Start sending data

AT+CIPSEND

>

10. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

11. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

12. Close the connection

AT+CIPCLOSE

Response:

CLOSED

OK

3.UDP Client Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=2 //Serial to WiFi mode
```

Response:

OK

2. Connect to a router

```
AT+CWJAP="SSID","password"
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:STA,"192.168.1.103"
```

```
+CIFSR:STAMAC,"24:0a:c4:2a:25:8c"
```

```
+CIFSR:ETHIP,"0.0.0.0"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Connect the computer and the module to the same router, and use a network debugging tool on PC to establish a UDP connection.

For example: IP: 192.168.1.101; port: 8080

5. The module is connected to the server as a UDP client.

```
AT+CIPSTART="UDPC","192.168.1.101",8080,3333
```

Response:

OK

6. The module sends data to the server.

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:

```
Recv 4 bytes
```

```
SEND OK
```

7. Receive the server data

Response:

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

8.Enable the transparent transmission mode

```
AT+CIPMODE=1//
```

Response:

OK

9. Start sending data

AT+CIPSEND

>

10. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

11. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

12. Close the TCP connection

AT+CIPCLOSE

Response:

CLOSED

OK

4.UDP Server Connection

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=2 //Serial to WiFi mode
```

Response:

OK

2. Connect to a router

```
AT+CWJAP="SSID","password"
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:STA,"192.168.1.103"
```

```
+CIFSR:STAMAC,"24:0a:c4:2a:25:8c"
```

```
+CIFSR:ETHIP,"0.0.0.0"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Connect the computer and the module to the same router, and use a network debugging tool on PC to establish a UDP connection.

For example: IP:192.168.1.101; port: 8080

5. The module establishes a UDP connection as a UDP server

```
AT+CIPSTART="UDPS","192.168.1.101",8080,3333
```

Response:

OK

7. The module sends data to the client.

```
AT+CIPSEND=4 // set data length which will be sent, such as 4 bytes
```

```
>test // enter the data, no CR
```

Response:

```
Recv 4 bytes
```

```
SEND OK
```

8. Receive the server data

Response:

```
+IPD,n:xxxxxxxx // received n bytes, data=xxxxxxxx
```

9. Enable the transparent transmission mode

```
AT+CIPMODE=1//
```

Response:

OK

10. Start sending data

AT+CIPSEND

>

11. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized, the transparent transmission will be exited.

12. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

13. Close the TCP connection

AT+CIPCLOSE

Response:

CLOSED

OK

HTTP Request

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=2 //Serial to WiFi mode
```

Response:

OK

2. Connect to a router

```
AT+CWJAP="SSID","password"
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:STA,"192.168.1.103"
```

```
+CIFSR:STAMAC,"24:0a:c4:2a:25:8c"
```

```
+CIFSR:ETHIP,"0.0.0.0"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Connect the computer and the module to the same router.

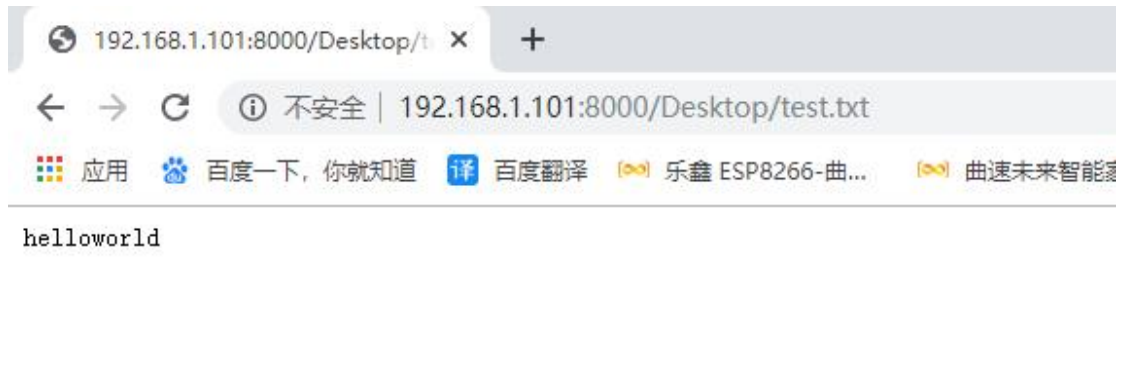
5. Set up a HTTP server on the PC



```
C:\WINDOWS\system32\cmd.exe - python -m SimpleHTTPServer 8000
D:\Users\Administrator>
D:\Users\Administrator>
D:\Users\Administrator>python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
```

6. Test an HTTP request on Web page





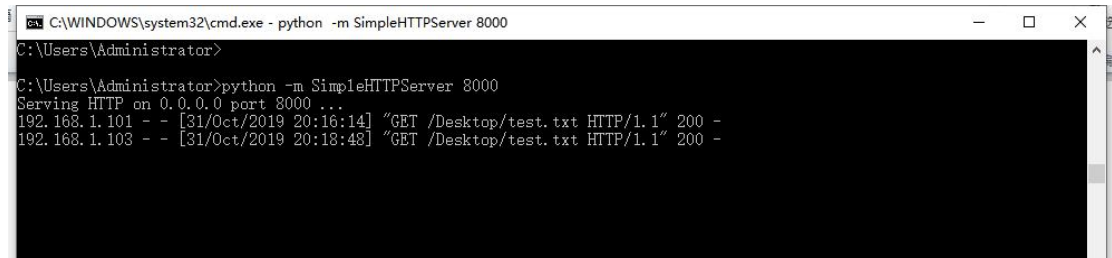
7. The module sends an HTTP request.

AT+CIPSTART="HTPC","http://192.168.1.101:8000/Desktop/test.txt"

Response:

OK

+IPD,12:helloworld



Serial to Bluetooth Transparent Data Transmission

1. Configure the pass-through channel of the module

AT+PASSCHANNEL=3 //Serial to Bluetooth mode

Response:

OK

2. Restart the module to enter the serial to Bluetooth transparent transmission mode

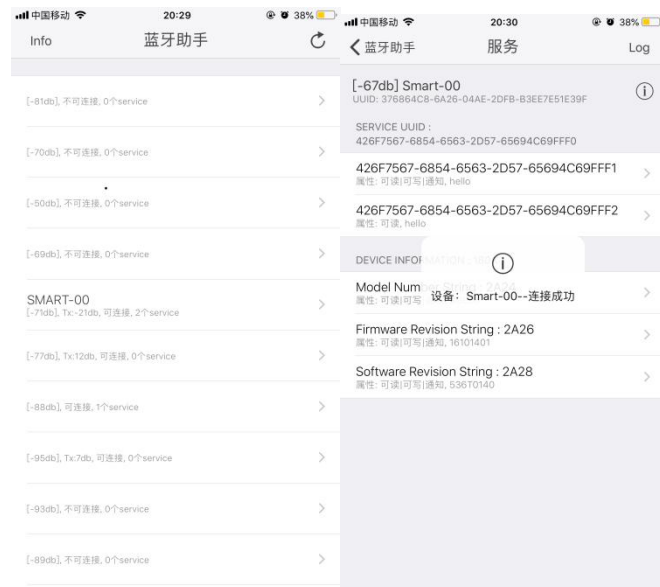
AT+RST

Response:

OK

3. Establish a Bluetooth connection

Establish a Bluetooth connection using a mobile Bluetooth tool app



4. Enable the transparent transmission mode

AT+CIPMODE=1//

Response:

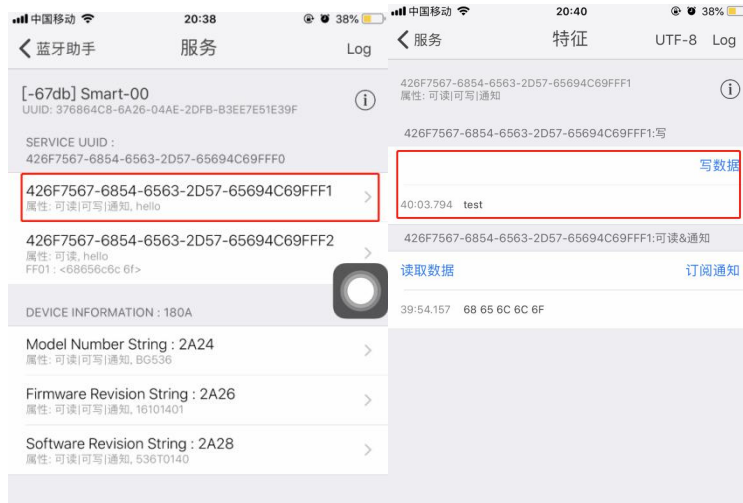
OK

5. Start sending data

AT+CIPSEND

>

6. The App sends Bluetooth data.



7. Receive Bluetooth data

Response:

+BLED,4:test

8. Exit sending data

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

9. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

Bluetooth to WiFi Transparent Data Transmission

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=5 //Bluetooth to WiFi transparent transmission mode
```

Response:

OK

2. Restart the module to enter Bluetooth to WiFi transparent transmission mode

```
AT+RST
```

Response:

OK

2. Connect to a router

```
AT+CWJAP="SSID","password"
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:STA,"192.168.1.103"
```

```
+CIFSR:STAMAC,"24:0a:c4:2a:25:8c"
```

```
+CIFSR:ETHIP,"0.0.0.0"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Connect the computer and the module to the same router, and use a network debugging tool on PC to establish a TCP server

For example: IP: 192.168.1.101; port: 8888

5. The module is connected to the server as a TCP client.

```
AT+CIPSTART="TCP","192.168.1.101",8888//protocol, server IP and port
```

Response:

OK

6. Enable the transparent transmission mode

```
AT+CIPMODE=1//
```

Response:

OK

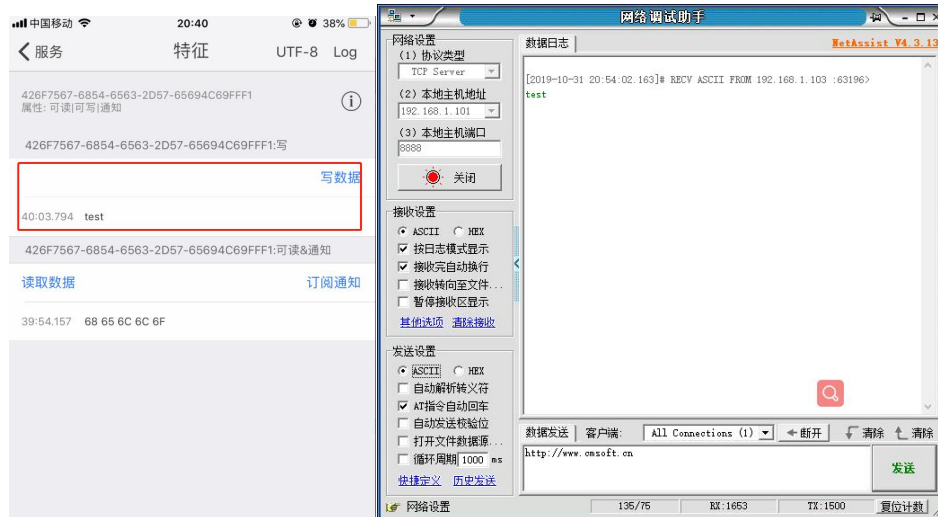
7. Start sending data

```
AT+CIPSEND
```

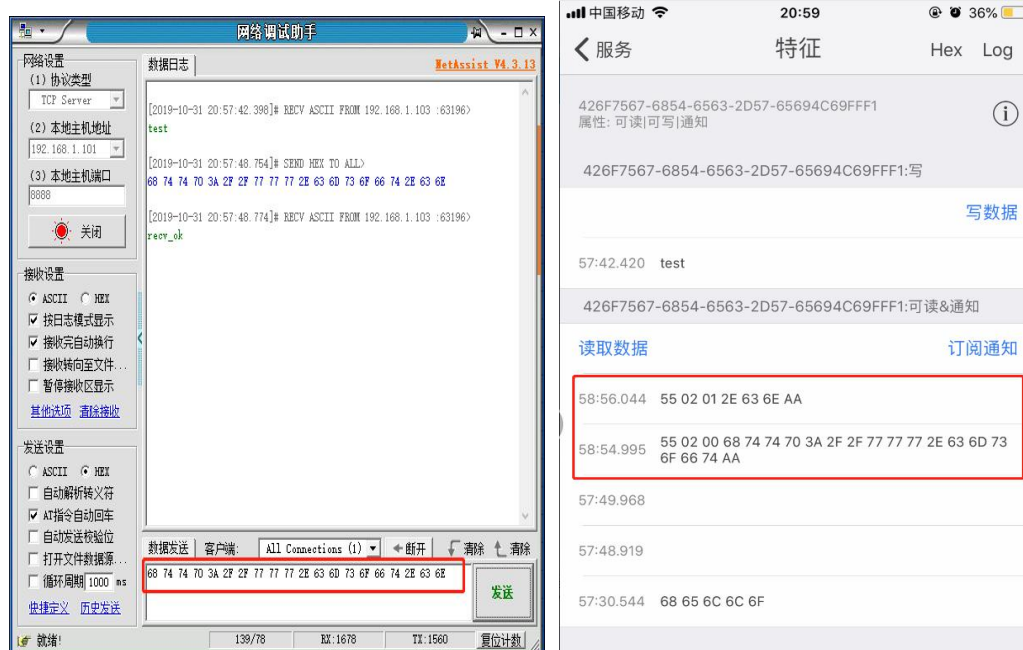
>

8. Bluetooth and TCP transparent data transmission

Bluetooth data→TCP server



TCP server data→Bluetooth



9. Exit sending data

(Bluetooth sending +++)

In the process of transparent transmitting, if a single packet of data "+++" is recognized, the transparent transmission will be exited.

10. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK

Note: Bluetooth to Ethernet transparent transmission is similar to Bluetooth to wifi transparent transmission

WiFi to Ethernet Transparent Data Transmission

1. Configure the pass-through channel of the module

```
AT+PASSCHANNEL=4 //WiFi to Ethernet transparent transmission mode
```

Response:

OK

2. Restart the module to enter WiFi to Ethernet transparent transmission mode

```
AT+RST
```

Response:

OK

3. Query the module's IP address

```
AT+CIFSR
```

Response:

```
+CIFSR:APIP,"192.168.4.1"
```

```
+CIFSR:APMAC,"24:0a:c4:2a:25:8d"
```

```
+CIFSR:ETHIP,"192.168.1.102"
```

```
+CIFSR:ETHMAC,"24:0a:c4:2a:25:8f"
```

OK

4. Connect the module to the same router, and use a network debugging tool on PC to create a TCP server.

For example: IP: 192.168.1.101; port: 8888

5. The module is connected to the server as a TCP client.

```
AT+CIPSTART="TCPC","192.168.1.101",8888//protocol, server IP and port
```

Response:

OK

6. Enable the transparent transmission mode

```
AT+CIPMODE=1//
```

Response:

OK

7. Start sending data

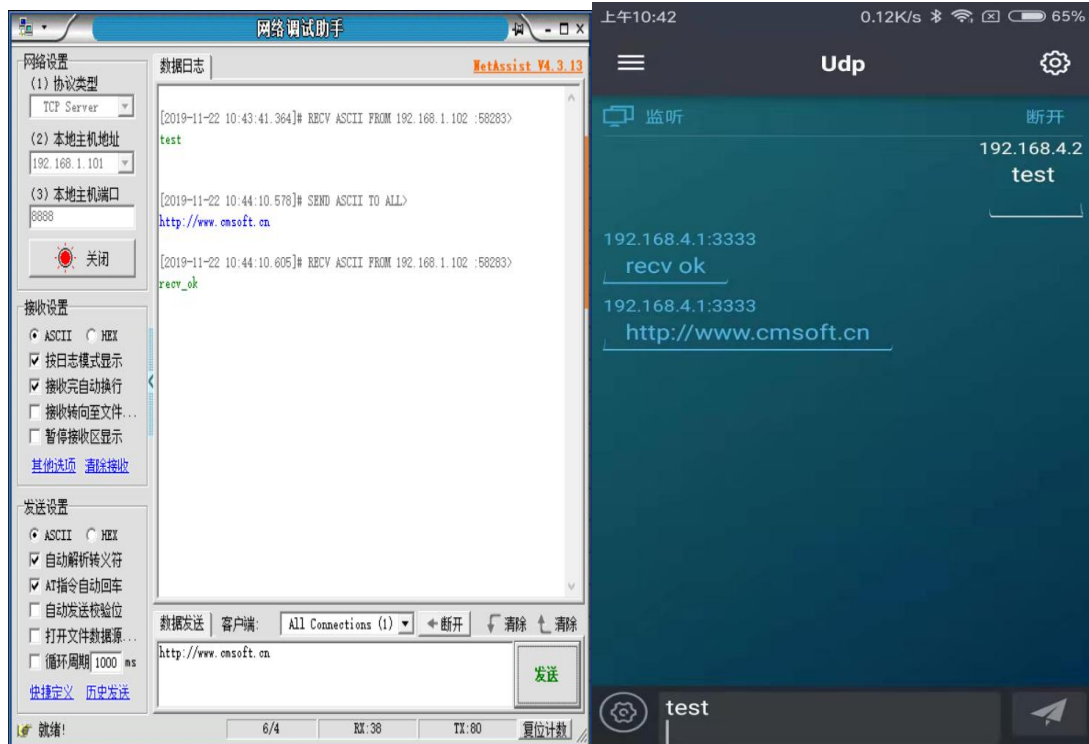
```
AT+CIPSEND
```

>

8. The mobile phone is connected to the hot spot issued by the module ssid: WT32-ETH01; password: 12345678

Create a UDP connection to the module using a mobile network debugging assistant

Module IP:192.168.4.1; Port number:3333



9. Exit sending data

(Send by phone+++)

In the process of transparent transmitting, if a single packet of data "+++" is recognized , the transparent transmission will be exited.

10. Exit the transparent transmission mode

AT+CIPMODE=0

Response:

OK