

## Chapter 18: Raspberry pi socket communication method

TCP/IP is the basis of the http protocol, and socket is the implementing of the TCP/IP. A socket has at least two endpoints, a server and a client client.

### Socket features:

- 1)The connection provided by the TCP/IP socket is reliable.
- 2)The socket is bidirectional.
- 3)Sockets can be one-to-many.

Because Raspberry Pi is a highly integrated micro-board with Wi-Fi module and support Python, so socket is a suitable method for network communication with other devices.

### Several important functions of Python socket:

- 1) `socket.bind(address)` it is used to bind the address. The address consists of the IP address and the port number set for this connection. When both server programs and client programs are in the machine, the IP address is '127.0. 0.1'.
- 2) `socket.listen(backlog)` it be used to set the maximum number of connections allowed to be bound.
- 3) `socket.accept()` allows the connection, when the server proceeds to this step, it represents the server is waiting for the client's connection;
- 4) `socket.connect(address)` The client sends a connection request;
- 5) `socket.recv(bufsize[,flag])` The message "receives" and accepts the data of the socket. The data is returned as a string, and bufsize is the maximum number that can be received, typically set to 1024 or 2048. Flag provides additional information about the message and it can be ignored.
- 6) `socket.send(string[,flag])` The message "send", the data in the string is sent to the connected socket. The return value is the number of bytes to send, which may be less than the byte size of string.
- 7) `socket.close()` closes the socket.

We need to complete the network communication between the Raspberry Pi and the PC by the network debugging tool. We need to choose the right PC software. The window system can choose NetAssist.

The Mac system also has many similar network debugging assistants.

The source code of the program is located at: </home/pi/yahboom/socket/server.py>

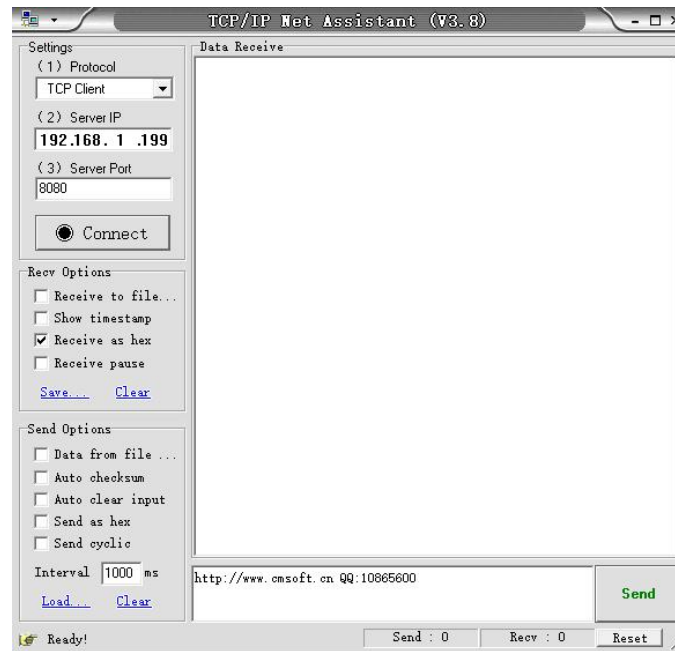


Figure 1-1 Windows NetAssist.exe

**Situation 1:**

Raspberry Pi as a server and the PC side as a client.

```

1  #!/usr/bin/env python2
2  # -*- coding: utf-8 -*-
3  """
4      * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
5      * @file      socket server
6      * @version    V1.0
7      * @details
8      * @par History
9
10     @author: longfuSun
11 """
12 import socket
13 import threading
14 #Set mode of socket is tcp/ip
15 s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
16 #You can modify the port according to the actual situation.
17 #address is Raspberryn Pi IP
18 address='192.168.1.157'      #wifi
19 port =9902
20 #Bind address and port
21 s.bind((address,port))
22 #Set the number of servers allowed to access
23 s.listen(2)a
24 sock,addr=s.accept()
25 true=True
26
27 #For avoid congestion,write receive into threads
28 def rec(sock):
29     global true
30     while true:
31         #Set encoding format utf-8
32         t=sock.recv(1024).decode('utf8')
33         #Exit when the input 'exit'
34         if t=='exit':
35             true=False
36         print('recieve: '+t)
37 trd=threading.Thread(target=rec,args=(sock,))
38 trd.start()
39 while true:
40     t=raw_input()
41     sock.send(t.encode('utf8'))
42     if t=='exit':
43         true=False
44 s.close()

```

Please enter the following command to run the program:

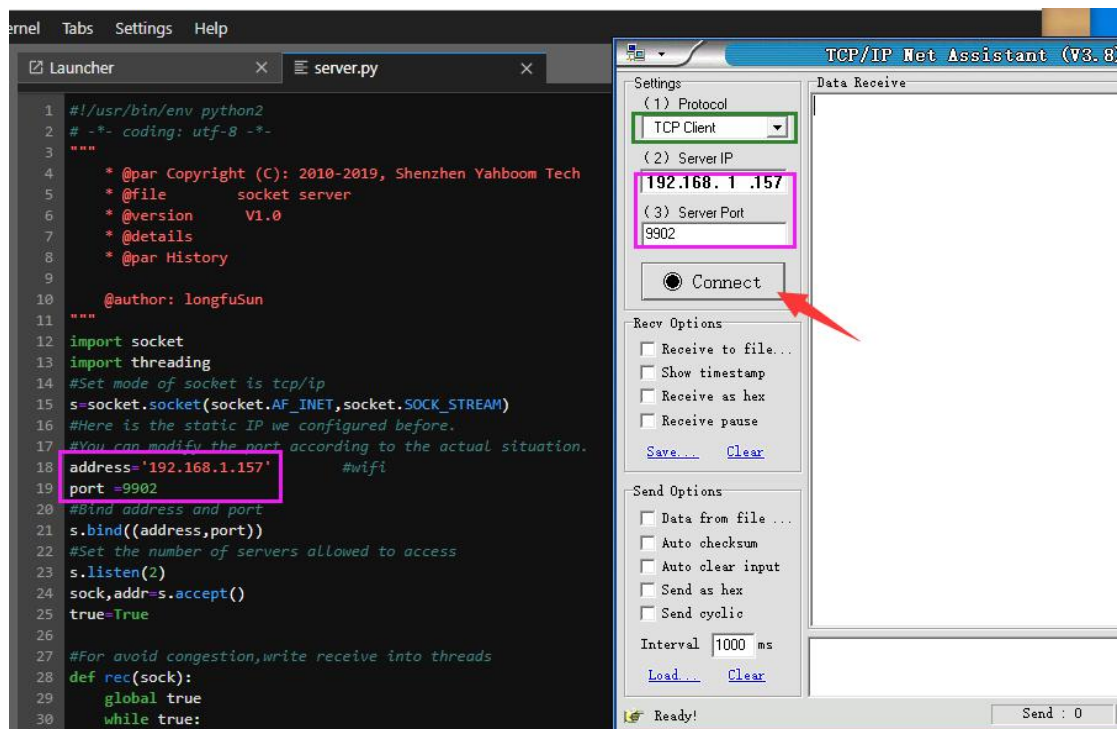
**python server.py**

**!!Note: This program cannot be run using Jupyter Lab.**

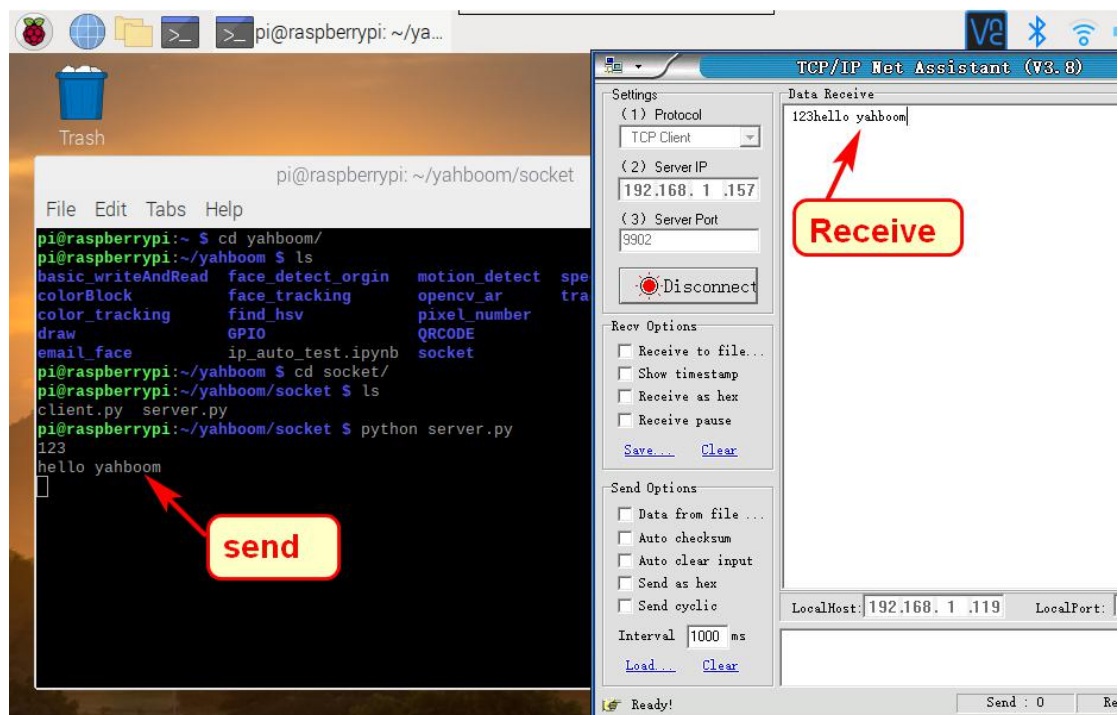
1-1 Because the Raspberry Pi is going to act as a server, so we need to run the above program.

1-2 We can know from the above code that Raspberry Pi declare local network address is '192.168.1.66', the port is: 9902.

Therefore, we need to configure this message on the Network Debugging Assistant, and click **【Connect】** . As shown below.



The result is as shown below.



## Situation 2:

Raspberry Pi as a client and the PC side as a server.

**Before run program, we need to start the server process on the PC.**

The source code of the program is located at: **/home/pi/yahboom/socket/clinet.py**

The Raspberry client program is as follows.

```

1  #!/usr/bin/env python2
2  # -*- coding: utf-8 -*-
3  """
4      * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
5      * @file      socket_client
6      * @version    V1.0
7      * @details
8      * @par History
9      * @author: longfuSun
10     """
11
12     import socket
13     import threading
14
15     #Set mode of socket is tcp/ip
16     #You can modify the port according to the actual situation.
17     #address is your PC IP
18     s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
19     s.connect(('192.168.1.119',9902)) #wifi
20     true=True
21     #For avoid congestion, write receive in the thread
22     def rec(s):
23         global true
24         while true:
25             t=s.recv(1024).decode('utf8')
26             if t=='exit':
27                 true=False
28             print('client received: '+t)
29     trd=threading.Thread(target=rec,args=(s,))
30
31     trd.start()
32     while true:
33
34         t=raw_input()
35         #send
36         s.send("client said: "+t.encode('utf8'))
37         if t=='exit':
38             true=False
39     s.close()
40

```

When the Raspberry Pi as a client, **the IP address in the program should be the IP address of the PC.**

Please modify it according to the your actual situation.

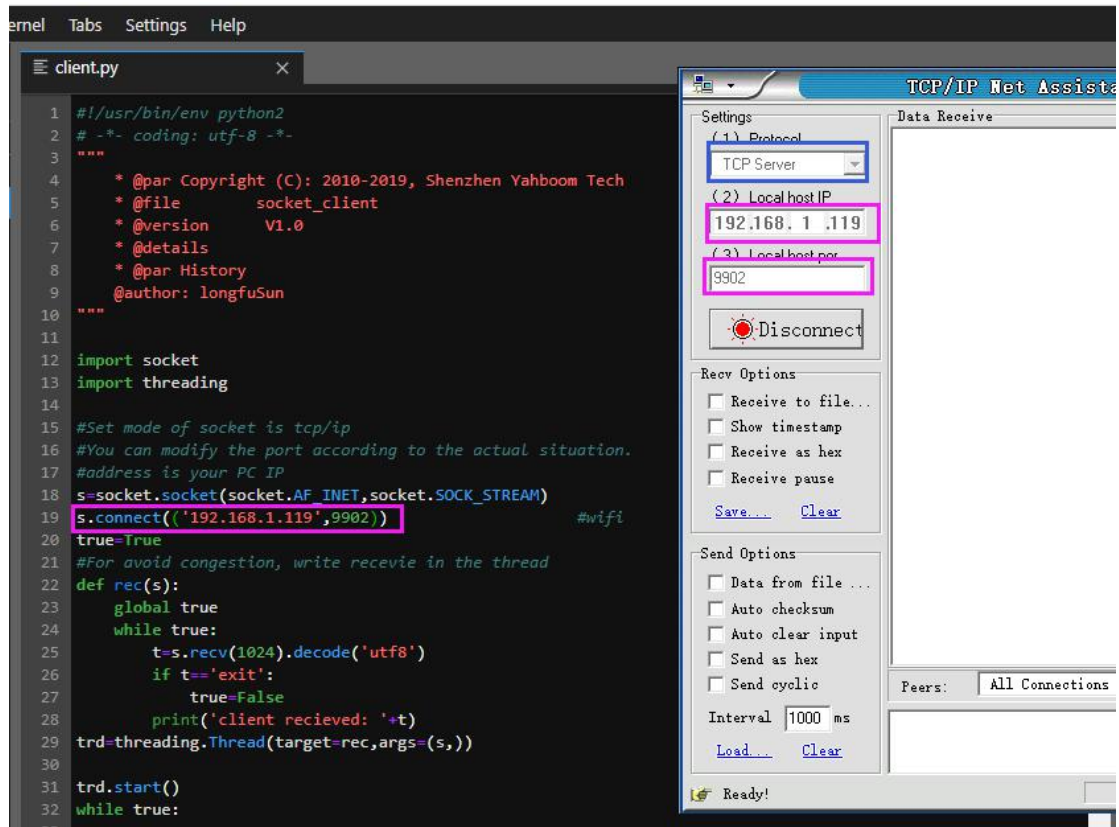
Please enter the following command to run the program:

**python client.py**

**!!Note: This program cannot be run using Jupyter Lab.**

We need to configure this message on the Network Debugging Assistant, and click **【Connect】** . As shown in below.





The result is as shown below.

