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 3B+ 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) socekt.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

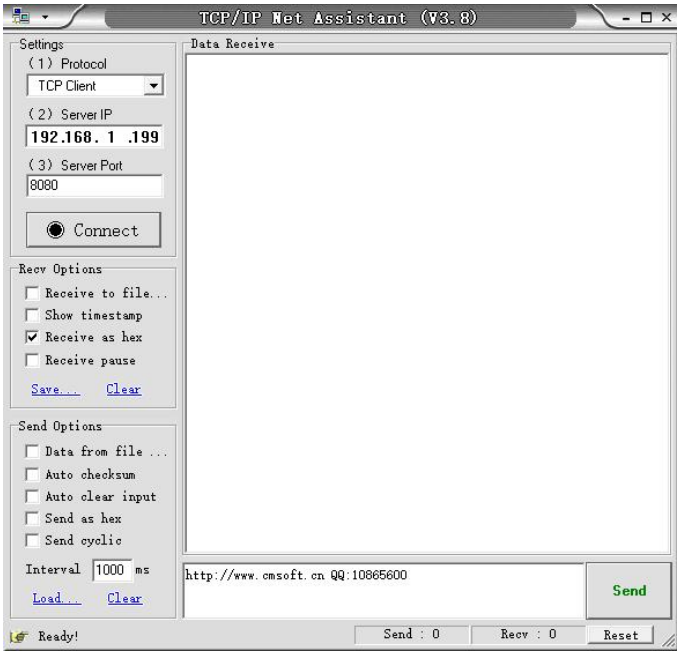

Figure 1-1 Windows NetAssist.exe

Situation 1:

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

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file        socket server
    * @version      V1.0
    * @details
    * @par History

    @author: longfuSun
"""
import socket
import threading
#Set mode of socket is tcp/ip
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
#Here is the static IP we configured before.
#You can modify the port according to the actual situation.
address='192.168.1.66'        #wifi
port =9902
#Bind address and port
s.bind((address,port))
#Set the number of servers allowed to access
s.listen(2)
sock,addr=s.accept()
true=True

#For avoid congestion,write receive into threads
def rec(sock):
    global true
    while true:
        #Set encoding format utf-8
        t=sock.recv(1024).decode('utf8')
        #Exit when the input 'exit'
        if t=='exit':
            true=False
        print('recieve: '+t)
trd=threading.Thread(target=rec,args=(sock,))
trd.start()
while true:
    t=raw_input()
    sock.send(t.encode('utf8'))

    if t=='exit':
        true=False
s.close()

```

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 in Figure 1-2 below.

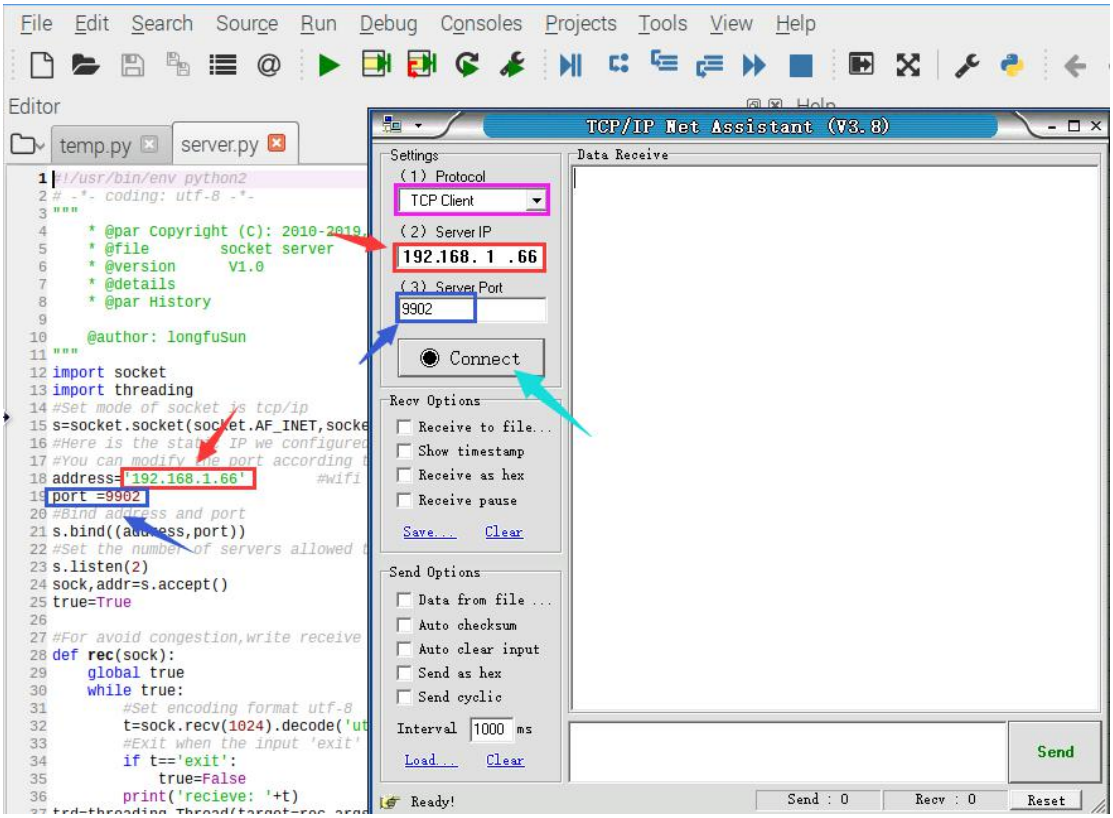

Figure 1-2

The result is as shown in the Figure 1-3 below:

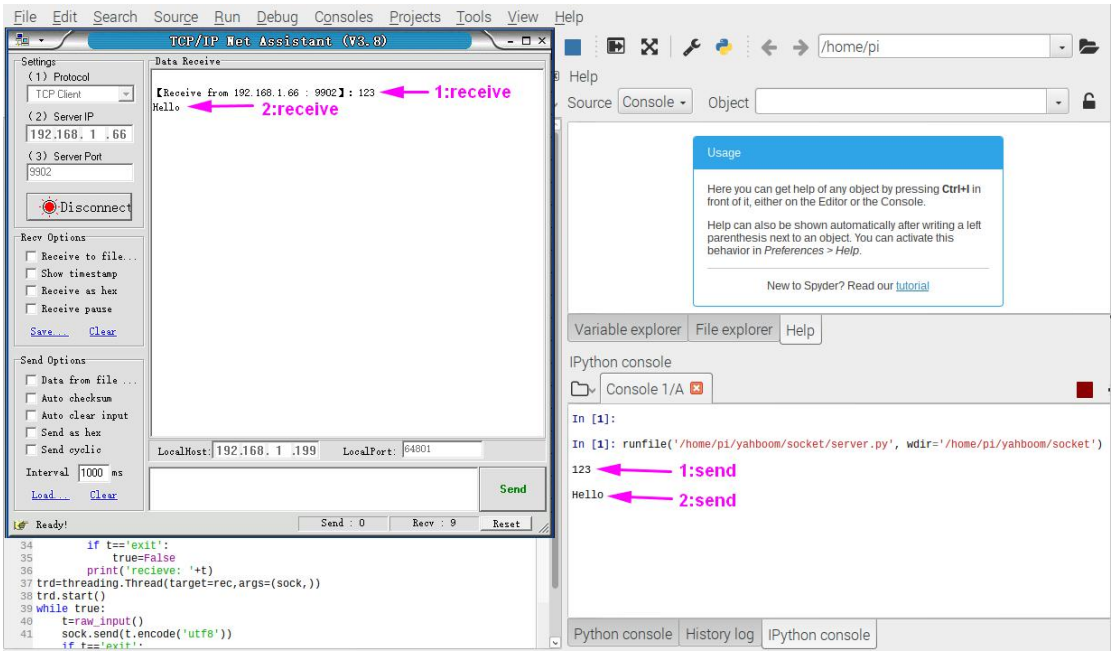

Figure 1-3-1

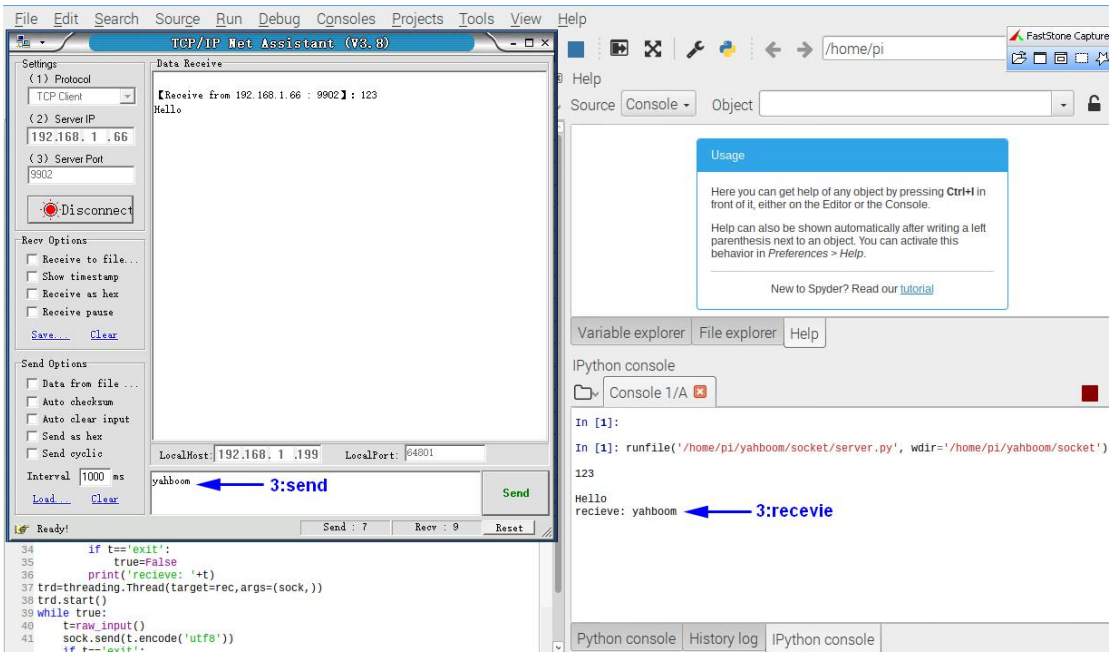

Figure 1-3-2

Situation 2:

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

The server process on the PC should be started first.

The source code of the program is located at:

/home/pi/yahboom/socket/clinet.py

The Raspberry client program is as follows.

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file        socket_client
    * @version      V1.0
    * @details
    * @par History
    @author: longfuSun
"""

import socket
import threading

#Set mode of socket is tcp/ip
#You can modify the port and IP address according to the actual situation.
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(('192.168.1.109',9999))                    #wifi
true=True
#For avoid congestion, write recevie in the thread
def rec(s):
    global true
    while true:
        t=s.recv(1024).decode('utf8')
        if t=='exit':
            true=False
        print('client recieved: '+t)
trd=threading.Thread(target=rec,args=(s,))

trd.start()
while true:

    t=raw_input()
    #send
    s.send("client said: "+t.encode('utf8'))
    if t=='exit':
        true=False
s.close()
```

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.

We need to configure this message on the Network Debugging Assistant, and click 【Connect】. As shown in Figure 1-4 below.

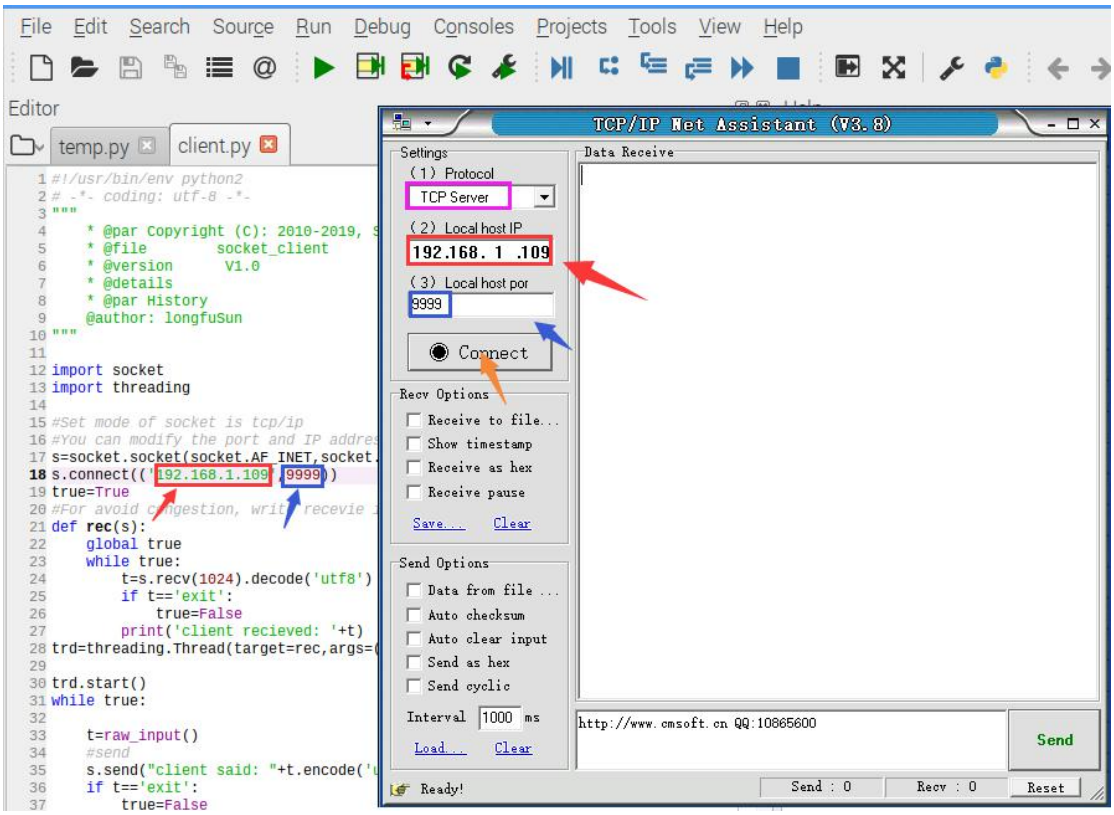
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor
temp.py
client.py
#!/usr/bin/env python2
-*- coding: utf-8 -*-
"""
* @par Copyright (C): 2010-2019,
* @file socket_client
* @version V1.0
* @details
* @par History
@author: longfuSun
"""
import socket
import threading
#Set mode of socket is tcp/ip
#You can modify the port and IP addres
s=socket.socket(socket.AF_INET,socket.
s.connect(('192.168.1.109',9999))
true=True
#For avoid congestion, write recevie
def rec(s):
global true
while true:
t=s.recv(1024).decode('utf8')
if t=='exit':
true=False
print('client recieved: '+t)
trd=threading.Thread(target=rec,args=(
trd.start()
while true:
t=raw_input()
#send
s.send("client said: "+t.encode('u
if t=='exit':
true=False
TCP/IP Net Assistant (V3.8)
Settings
(1) Protocol
TCP Server
(2) Local host IP
192.168. 1 .109
(3) Local host por
9999
Connect
Recv Options
Receive to file...
Show timestamp
Receive as hex
Receive pause
Save... Clear
Send Options
Data from file ...
Auto checksum
Auto clear input
Send as hex
Send cyclic
Interval 1000 ms
Load... Clear
Data Receive
http://www.cmsoft.cn QQ:10865600
Send
Ready!
Send : 0
Recv : 0
Reset

Figure 1-4