

DOP: / /2023

DOS: / / 2023

Experiment No: 09

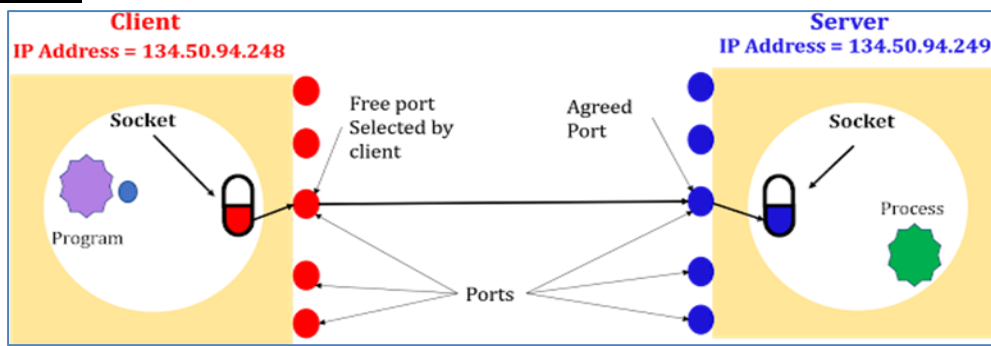
Aim: - Write a program to create TCP Server on Raspberry Pi and respond with humidity data to TCP client when requested.

Hardware/Software: Raspberry Pi

Theory:

A socket is one endpoint of a two-way communication link between two programs running on the network. The application creates a socket. Socket is an interface between application layer and transport layer. It is an interface (a "door") into which an application process can both send and receive message to/from another application process (remote/local application process). Socket is also referred as the application programmer's interface (API) between the application and the network. Need of Socket When we desire a communication between two applications possibly running on different machines, we need sockets. Requirement of socket is to build any Network Application i.e., Web browsers, FTP etc...

How Socket Works?

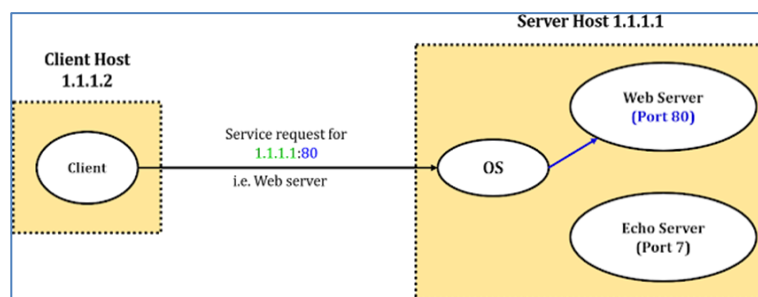


There are two different programs running on two different machines in network. Client program wants to communicate with server program. Client selects any free port from client machine and send data to server process. Server select port which is bind with server process. That port is called agreed or specific port. After that logical connection is created between client and server. Server received client process at agreed/specific port. Then communication is started between server and client.

What is Socket address?

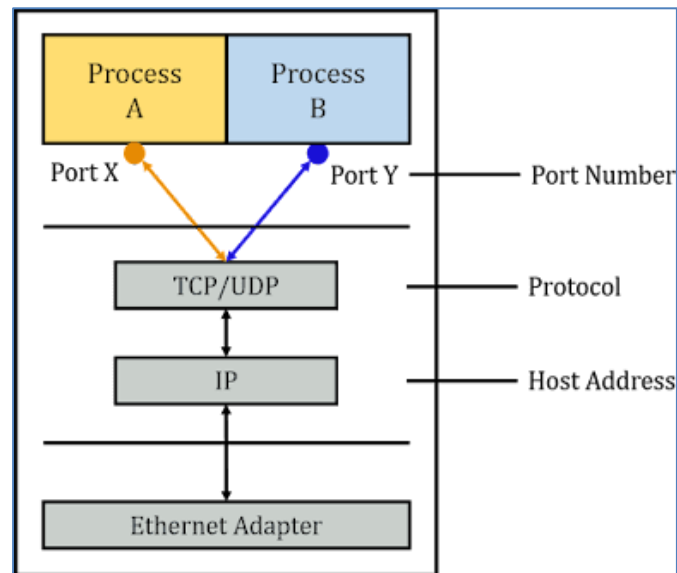
Socket is combination of IP address and Port number. IP address is used for host-to-host communication in network. IP address is assigned by network layer. Port number is used for process-to-process communication in network. Port number is assigned by transport layers protocol.

How process can identify using socket?



Let's take one example, there is one Server host and Client host in above diagram. There are two different servers are configured on server host. Client send a request message to server. Request message is received from port number 80. At server-side Web server is bind with port number 80. So, Server OS forward that request message to Web server. Process is identify based on port address. Port address is already assigned in unique socket address.

On which layer socket is execute in TCP/IP Model?



Types of Sockets:-

There are two types of sockets: Stream Socket and Datagram Socket.

1) **Stream Socket:-**

Stream socket is also called connection- oriented socket. It provides reliable, connected networking service. Error free; no out- of- order packets (uses TCP). Applications of Stream Sockets are telnet/SSH, http, https and many more.

2) **Datagram Socket:-**

Datagram socket is also known as Connectionless socket. It provides unreliable, best- effort networking service Packets may be lost; may arrive out of order (uses UDP). Application of Datagram Sockets are streaming audio/ video (real player) and many more. Raw Socket, Sequenced packet Socket etc.

Procedure:

Input:

****Simple Client Program****

```
#cd Desktop
```

```
#cd Sample_Codes_FEB23
```

```
#cd Socket_Examples
```

```
#python3 Simple_Server.py import socket
```

```
import threading import time
```

```
HOST = '127.0.0.1'
```

```
PORT = 65432
```

```
def process_data_from_server(x): x1, y1 = x.split(",")
```

```
return x1, y1 def my_client():
```

```
threading.Timer(11, my_client).start()
```

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:s.connect((HOST, PORT))
```

```
my = input ("Enter Command: ") # Data to be entered
```



```
my_inp = my.encode('utf-8') s.sendall(my_inp)
data = s.recv(1024).decode('utf-8')
x_temperature, y_humidity = process_data_from_server(data)

print("Temperature {}".format(x_temperature)) print("Humidity {}".format(y_humidity)) s.close()
time.sleep(5)
if name == " main ": try:
while True: my_client()
# If keyboard Interrupt (CTRL-C) is pressed except KeyboardInterrupt:
print ('Exiting Program')
*****end*****

***Simple Server program****
import socket import numpy as np import encodings
HOST = '127.0.0.1'
PORT = 65432
def random_data():
x1 = np.random.randint(0, 55, None)
y1 = np.random.randint(0, 75, None)
my_sensor = "{},{}".format(x1,y1)

return my_sensor
def my_server():
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s: print("Server Started waiting
for client to
    connect")
    s.bind((HOST, PORT))
s.listen(5)
conn, addr = s.accept()
with conn:
print('Connected By:',addr) while True:

data = conn.recv(1024).decode('utf-8') if str(data) == "Data":
print("OK Sending Data") my_data = random_data()
x_encoded_data = my_data.encode('utf-8') conn.sendall(x_encoded_data)

elseif str(data) == "Quit":
print ("shutting down server") break
if not data: break
else:
pass
if name == "main": try:
while True: my_server()
# If keyboard Interrupt (CTRL-C) is pressed except KeyboardInterrupt:
print ('Exiting Program')

*****end*****
```



Jawahar Education Society's Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

Result:

```
1 #include <DHT.h> // Including library for dht
2 #include <ESP8266WiFi.h>
3
4 const char *ssid = "Lavesh hotspot"; // replace with your wifi ssid and wpa2 key
5 const char *pass = "RightWing";
6 #define DHTPIN 2 //pin where the dht11 is connected
7 DHT dht(DHTPIN, DHT11);
8 WiFiClient client;
9 void setup()
10 {
11   Serial.begin(9600);
12   delay(10);
13   dht.begin();
14   Serial.println("Connecting to ");
15   Serial.println(ssid);
16   WiFi.begin(ssid, pass);
17   while (WiFi.status() != WL_CONNECTED)
18   {
19     delay(500);
20     Serial.print(".");
21   }
22   Serial.println("");
23   Serial.println("WiFi connected");
24 }
25 void loop()
26 {
27   float h = dht.readHumidity();
28   float t = dht.readTemperature();
29   // Print the temperature and humidity values
30   Serial.print("Temperature:");
31   Serial.print(t);
32   Serial.print(" Humidity:");
33   Serial.print(h);
34   Serial.println();
35   delay(1000);
36 }
```

Executing the TCP server with client and receiver

```
1 import Adafruit_DHT as dht
2 import RPi.GPIO as gpio
3 from time import sleep
4 sensor=dht.DHT11
5 import sys
6 gpio=4
7 from urllib.request import urlopen
8 myAPI= '2Y9MJ3F3APTNNIM'
9 baseURL = 'https://api.thingspeak.com/update?api_key='
10
11 while True:
12     hum, temp=dht.read_retry(sensor, gpio)
13     if hum>0 and temp>0:
14         HEADER = 'fields={}&field2={}'.format(temp,hum)
15         New_URL=baseURL + myAPI + HEADER
16         print ("Temperature = {} Humidity = {}".format(temp,hum))
17         conn=urlopen(New_URL)
18         sleep(2)
19     else:
20         print("Failed to get reading. Try again")
```

Sending forward the measures of temperature and humidity through TCP servers.



Got the measures of the server for temperature and Humidity through the sensors of Raspberry PI.

Conclusion:

Thus, we were able to tell the client respective temperature and humidity by creating a TCP server on raspberry PI.