4. Read data from sensor and send it to a requesting client. (using socket communication)

→ Here, we are mainly sending data from Raspberry pi (sensor) to client using socket get data.

→ Here the client reads temperature/humidity sensor data.

Code

Server side

```
import socket
import numpy as np
import encodings //to encode data
HOST = socket.gethostname( )
PORT = 65432
def random_data( ):
    pin = 4
    sensor = Adafruit_DHT.DHT22
    humidity, temperature = Adafruit.DHT.read_retry
                            (sensor, pin)
    'if humidity is not None and temperature
                is not None:
        print ('Temp = {0:0.1f}° C Humidity
        = {1:0.1f} %.' .format (temperature,
                                humidity))
        data = '{}{}'.format(temperature)humidity
        return data
```

```
def my_server():
    with socket.socket(socket.AF_INET, socket.
            sock_STRFAM) 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)
            elif str(data) == "quit":
                print("shutting")
                break
            if not data:
                break
            else:
                pass
if __name__ == "__main__":
    while 1:
        my_server()
```

client side :-

```
import socket
import threading
import time
HOST = socket.gethostname()
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:")
        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 %s".format
                                    (x_temperature))
        print("Humidity %s".format
                                    (y_humidity))
        s.close()
        time.sleep(5)
```

```
if __name__=="__main__":
    while 1:
        my_client()
```

Output:-

Server side:-

Server started waiting for client to connect
Connected by ('192.168.0.102', 50200)
ok sending data
Temp = 26.0° c Humidity = 28.0 %

client side :-

Enter command: Data
Temperature 26.0
Humidity 28.0