22/3/25

WEEK-9

Aim: Write a program to create a TCP server on cloud using Arduino and respond with humidity data to the TCP client when requested.

Software Required: ThingSpeak server, Arduino IDE

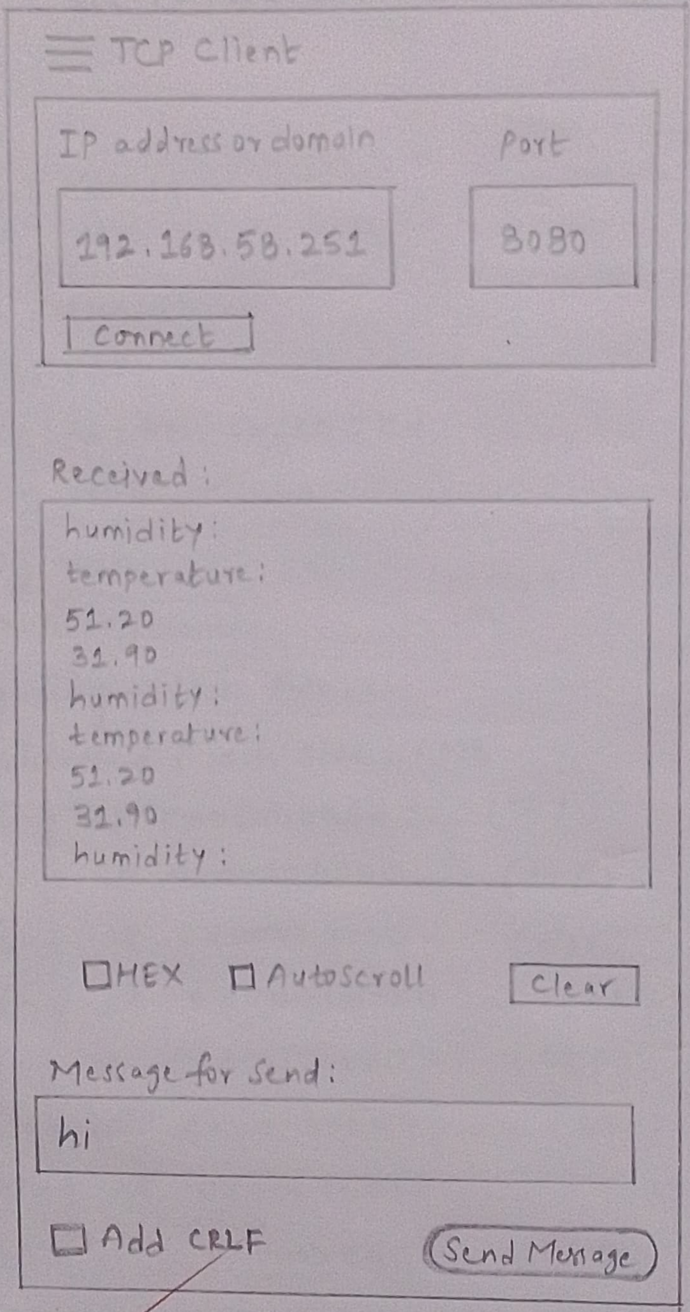
Source Code

```
#include "ESP8266WiFi.h"
#include "DHT.h"
const char* ssid = "Data";
const char* password = "12345678";
WiFiServer wifiserver(8080);
DHT dht (D3, DHT22);
void setup() {
        serial.begin (115200);
        delay (1000);
        WiFi.begin (ssid, password);
        while (WiFi.status() != WL CONNECTED) {
              delay (1000);
              serial.println ("Connecting... ");
        }
        serial.print ("Connected to wiFi. IP: ");
        serial.println (WiFi.localIP());
        wifiServer.begin();
        dht.begin();
void loop() {
      WiFiClient client = wifiserver.available();
      if (client) {
            while (client.connected()) {
                  while (client.available() > 0) {
                        float t=dht.readTemperature();
                        float h=dht.readHumidity();
                        client.print("humidity: ");
                        client.print ("temperature: ");
                        client.println(h);
                        client.println(h);
                        client.println (t);
                        client.println (t);
                        delay (2000);
            }
        }
        client.stop();
        Serial.println("Client Disconnected");
    }
}
```

OUTPUT:

```
Serial port COM19
Connecting...
Chip is ESP8266EX
Features: WiFi
MAC: ec:fa:bc:c2:cc:cd
uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Writing at 0x00000000... (7%)
Writing at 0x00004000... (15%)
Writing at 0x00008000... (23%)
Writing at 0x00014000... (46%)
Writing at 0x00018000... (53%)
Writing at 0x00020000... (69%)
Writing at 0x00024000... (76%)
Writing at 0x0002C000... (92%)
Writing at 0x00030000... (100%)
Wrote 283184 bytes (207814 compressed) at 0x00000000
  in 18.5 seconds (effective 122.5 kbits/s)...
Hash of data verified.
Leaving...
Hard resulting via RTS pin...
```

```
Connecting...
Connecting...
Connecting...
Connected to WiFi. IP: 192.168.58.251
51.20
31.40
52.70
31.10
52.70
31.10
52.70
31.10
52.70
31.10
53.00
31.10
53.50
31.00
53.50
31.00
53.50
31.00
53.30
31.00
```



IP address or domain: 192.168.58.251

Port: 8080

Received:

```
humidity:
temperature:
51.20
31.90
humidity:
temperature:
51.20
31.90
humidity:
```

Message for send: hi

12/4/25

12-04-2025 WEEK-10

Aim: write a program to create UDP server on cloud using Arduino and Respond with humidity data to UDP client when requested.

Hardware Requirements:

- Arduino UNO board
- Node MCU ESP 8266
- DHT11 (blue color) temperature and Humidity sensor
- Female Jumper wires

Procedure:

- Take DHT11 sensor connect positive to 3V neg. to GND and data to D3 or D4 according to the code.
- Download the application UDP terminal in Mobile.
- Change the ssid, password in the code and for UDP IP address - open UDP terminal app and goto setting > UDP settings > It shows Remote IP address of UDP.
- Verify the code in Arduino and in UDP terminal click on three dots > click on start.
- After compiling > upload, serial monitor shows the output.

Source Code:

```
#include <ESP8266 WiFi.h>
#include <WiFiUdp.h
#include <DHT.h>
const char* ssid = "hello";
const char* password = "123456789";
const char* udpAddress = "192.168.0.7";
const int udpPort = 1234;
#define DHTPIN D3
```

```
#define DHTTYPE DHT11
DHT dht (DHTPIN, DHTTYPE);
WiFiUDP udp;
void setup () {
    Serial.begin (115200);
    Serial.println ();
    Serial.println (" Connecting to WiFi... ");
    WiFi.begin (ssid, password);
    while ( wifi.status () ! = WL_CONNECTED) {
        delay (1000);
        serial.print (" connecting ");
    }
    Serial.println ();
    serial.print ('connected to WiFi. IP: ");
    dht.begin ();
}
void loop () {
    delay (10000);
    float temperature = dht.readTemperature ();
    float humidity = dht.readHumidity ();
    if (isnan(temperature) || isnan (humidity)) {
        Serial.println ("Failed to read from DHT
                                         sensor ");
        return;
    }
    Serial.print ("Temperature ");
    serial.print (temperature);
    Serial.print ("°C \t Humidity : ");
    Serial.print (humidity);
    serial.println ("% ");
    Serial.println ("Sending data over UDP...");
    udp.beginPacket (udpAddress, udpPort);
    udp.print ("Temperature: ");
    udp.print (temperature);
    udp.print (" °c, Humidity: ");
```

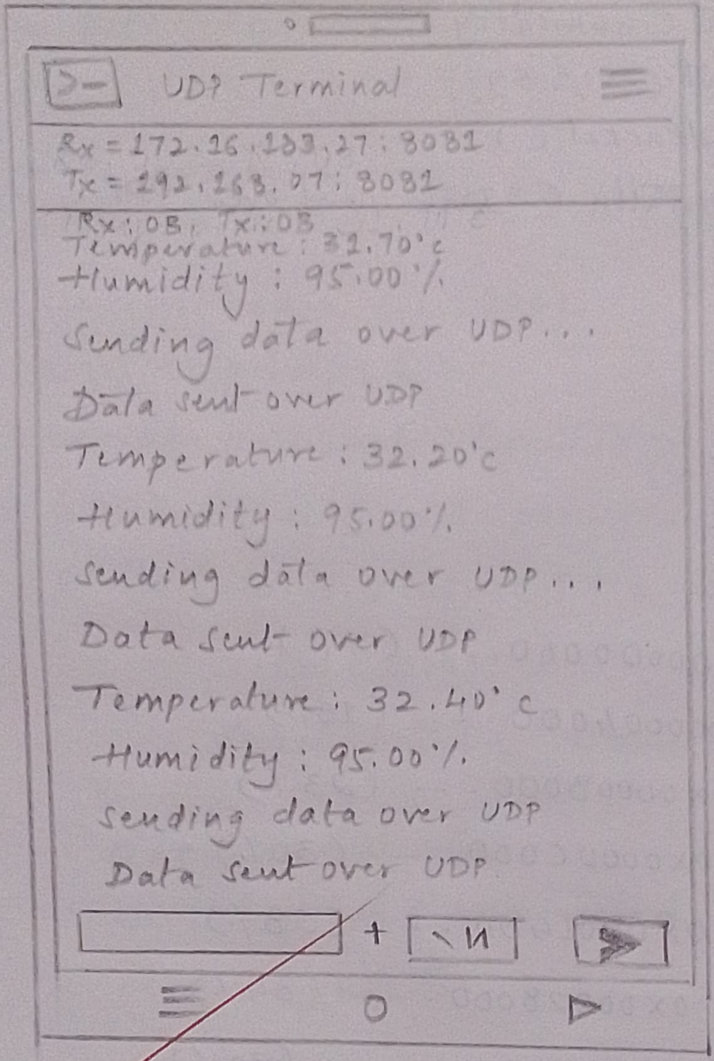
```
        udp.print(humidity);
        udp.print("%.");
        udp.endPacket();
        Serial.println("Data sent over UDP.");

}
```

Output:

```
uploading stub...
Running stub...
stub running...
Writing at 0x00000000... (7%)
Writing at 0x0004000... (15%)
Writing at 0x00008000... (23%)
Writing at 0x0000C000... (30%)
Writing at 0x0001400... (38%)
Writing at 0x00028000... (84%)
Writing at 0x0002C000... (92%)
Writing at 0x00030000... (100%)
Wrote 282912 (207366 compressed) at 0x00000000
in 18.3 seconds.
```

```
Serial Monitor
Temperature : 31.90° C  Humidity : 95.00%.
sending data over UDP..
Data sent over UDP.
Temperature : 32.20°C  Humidity : 95.00%.
sending data over UDP...
Data sent over UDP
Temperature : 32.40°c  Humidity : 95.00%.
Sending data over UDP...
Data sent over UDP
Temperature : 32.30°c  Humidity : 95.00%.
sending data over UDP..
Data sent over UDP.
```



```
Rx = 172.16.183.27 : 8081
Tx = 192.168.77 ; 8082
Rx : 0B, Tx : 0B
Temperature : 31.70°c
Humidity : 95.00%
Sending data over UDP...
Data sent over UDP
Temperature : 32.20°c
Humidity : 95.00%
Sending data over UDP...
Data sent over UDP
Temperature : 32.40°c
Humidity : 95.00%
Sending data over UDP
Data sent over UDP
```

19/4/25