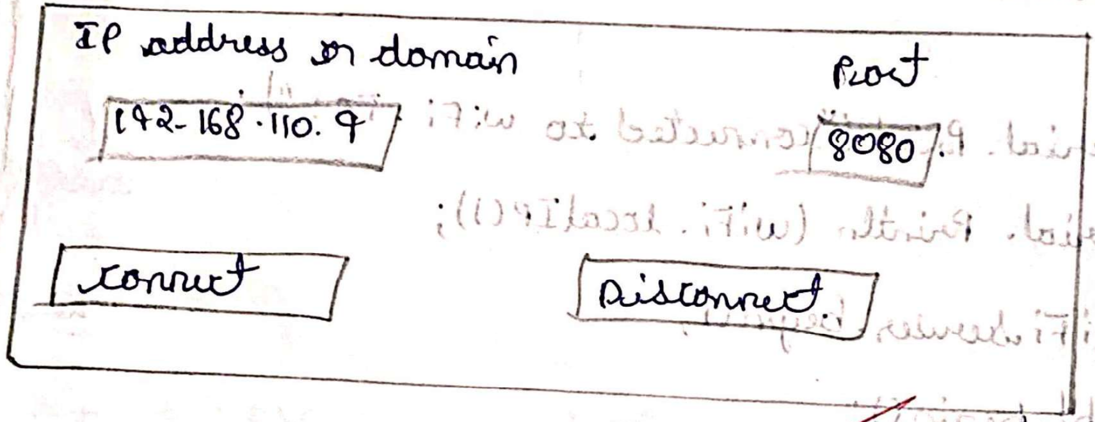
```
client. stop();
serial. println ("client disconnected");
 }
}
```

Output:



IP address or domain

192.168.110.9

Port

8080

Week - 10

Aim: Write a Program to create UDP Server on cloud using Arduino and Respond with humidity data to UDP client when requested.

Software Required.

- Thingspeak Server.
- Arduino IDE

Source code:

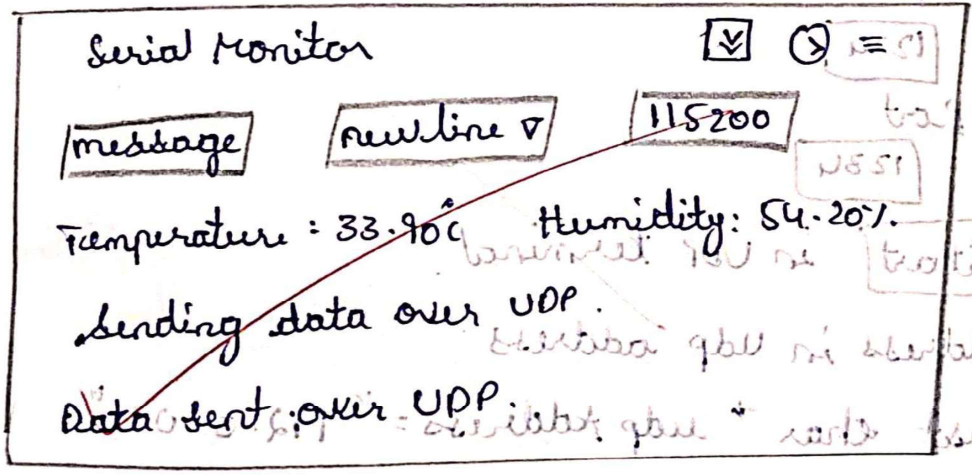
```
# include < ESP 8266 WiFi.h>
# include <WiFiUdP.h>
# include <DHT.h>
const char *ssid = "xxxx";
const char * Password = "xxxx";
const char * udpAddress = "xxxx";
const int udp port = 1234;
# define DHTPIN, D3
# define DHTTYPE DHT11
DHT dht (DHTPIN, DHTTYPE);
WiFiUDP udp;
Void setup () {
serial. begin (115200);
serial. Println ();
```

```
Serial. Println ("connecting to wiFi---");
WiFi. begin (ssid, Password);
while (wiFi. Status () != WL_CONNECTED) {
 delay (2000);
 Serial. Print (".");
}
 Serial. Println ();
 Serial. Println ("wiFi connected.");
 dht. begin ();
 }
Void loop () {
  delay (2000);
  float temperature = dht. readTemperature ();
  float humidity = dht. readHumidity ();
  if (isnan (temperature) || isnan (humidity)) {
  Serial. Println ("Failed to read from DHT sensor!");
  }
Serial. Print ("Temperature: ");
Serial. Print (temperature);
Serial. Print (" °c / 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.Println("%");
udp.endpacket();
Serial.Printeln("Data sent over UDP");

}.
```

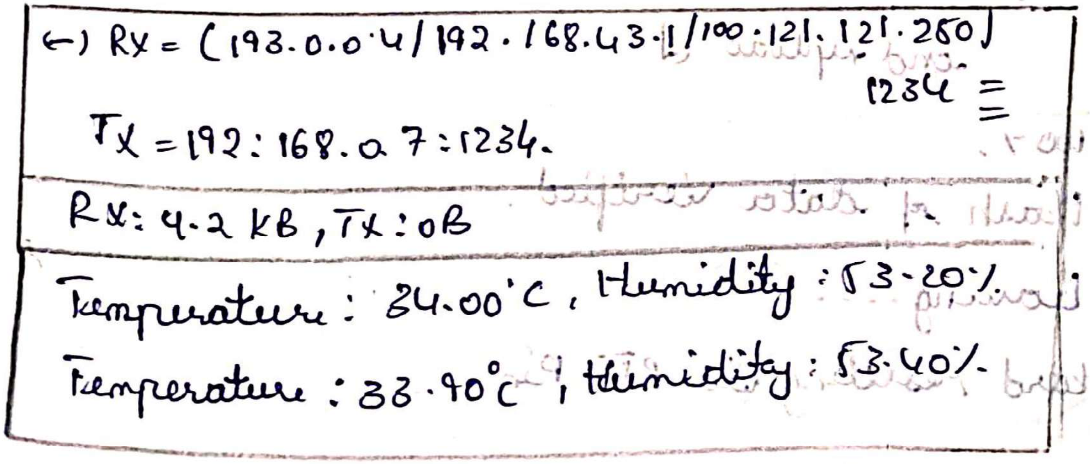
output:

In serial monitor.



```
Temperature : 33.90°c   Humidity: 54.20%.
Sending data over UDP.
Data sent over UDP.
```

In UDP terminal.



```
←) RX = (193.0.0.4/192.168.43.1/100:121.121.280)
                                            1234
   TX = 192:168.a 7:1234.
RX: 4.2 KB , TX: 0B
Temperature: 34.00°C, Humidity: 53.20%.
Temperature: 33.90°c , Humidity: 53.40%.
```

Procedure :

1) const char udp Address = "";

Keep it empty.

compile ↵

and upload ↵

```
100%.
Hash of data Verified
Leaving....
Hard resetting via RTS Pin...
```

* In Udp terminal

↳ setting.
↳ UDP Settings.
↳ Remote Port

1234 ○ ☒

↳ local Port

1234

2) Now Start in UDP terminal.

3) Rx address in udp address

const char * udp Address = "192.0.0.4";

again compile ↵

and upload ↵

```
100%.
Hash of data Verified.
Leaving....
Hard resetting via RTS Pin...
```

4) open Serial monitor.
you will get.

```
connecting to wiFi
connected...
Data is send over.
```