

Week 7.

Aim: Write a Program to upload temperature and Humidity data to the cloud using an Arduino.

Hardware Requirements.

1. Arduino Uno board
2. NodeMCU ESP 8266 Breakout Board.
3. DHT-11/DHT-22 Temperature and Humidity Sensor.
4. Jumper wires.
5. Bread board.
6. WIFI Network.

Source Code:

```
#include <DHT.h>
#include <ESP8266WiFi.h>

String apiKey = "WN1Y86N7XXAFGUP";
const char* ssid = "Samsung A12";
const char* Pass = "xxxxxx";
const char* server = "api.thingspeak.com";

#define DHTPIN D3
DHT dht(DHTPIN, DHT11);
WiFiClient client;

void setup() {
  Serial.begin(115200);
  delay(1000);
```

```

dht.begin();
Serial.println("connecting to");
Serial.println(ssid);
WiFi.begin(ssid, Pass);
while (WiFi.status() != WL_CONNECTED)
{
    delay(2000);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (isnan(h) || isnan(t))
    {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
    if (client.connect(server, 80))
    {
        String postStr = apiKey;
        postStr += "&field1=";
        postStr += String(h);
    }
}

```


Poststr += "4 field 2=";

Poststr += String(t);

Poststr += "\r\n\r\n";

client.Print("Post/update HTTP/1.1\n");

client.Print("Host: api.thingspeak.com\n");

client.Print("connection: close\n");

client.Print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

client.Print("Content-Type: application/x-www-form-urlencoded\n");

client.Print("Content-length: ");

client.Print(Poststr.length());

client.Print("\n\n");

client.Print(Poststr);

Serial.Print("Temperature: ");

Serial.Print(t);

Serial.Print("degree Celsius, Humidity: ");

Serial.Print(h);

Serial.Println("Y... send to Thingspeak:");

{

client.stop();

Serial.Println("waiting ...");

delay(1000);

}

output:

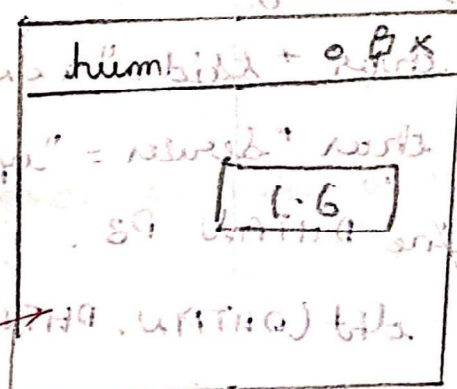
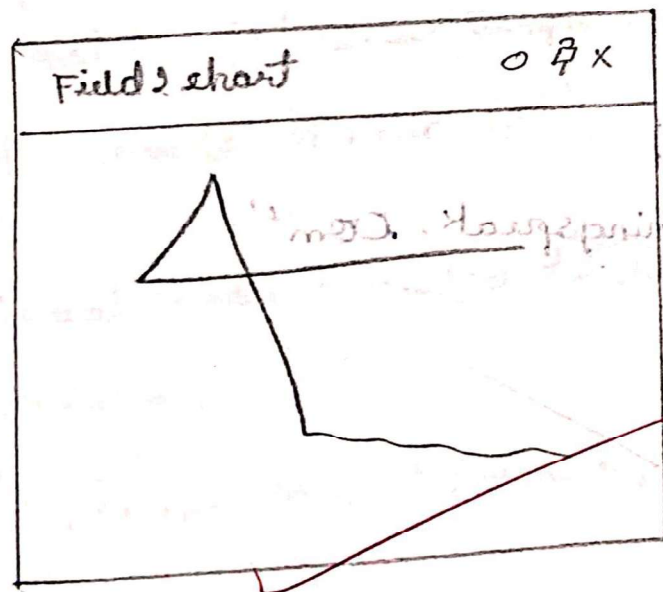
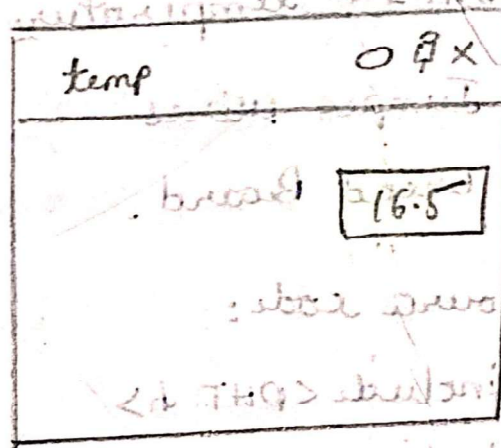
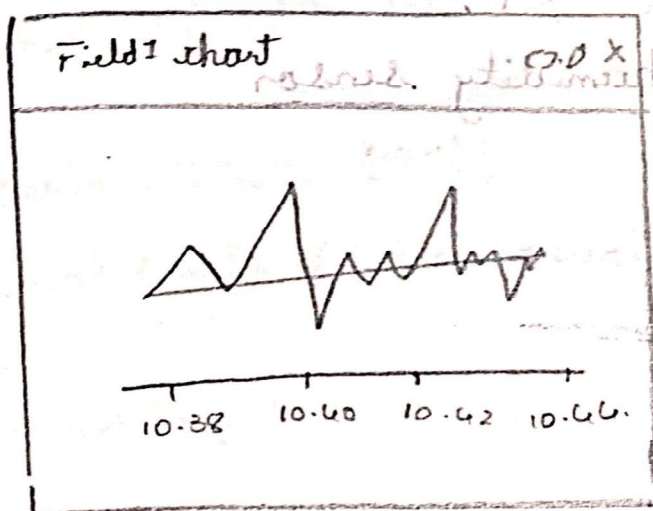
Thingspeak channels ▽ Apps ▽ Devices ▽ Support ▽

Temperature sensor

channel ID: 2467178.

Access: Private.

Private View.



bleed