



### Experiment -1

|                 |                          |          |                             |
|-----------------|--------------------------|----------|-----------------------------|
| Programme       | : B.Tech. CSE            | Semester | : Winter 2023-24            |
| Course Title    | : Internet of Things Lab | Code     | : BECE352E                  |
|                 |                          | Slot     | : L23-24                    |
| Register Number | : 21BAI1106              | Name     | : Ojas Patil                |
| Faculty (s)     | : Dr. Manimaran P        | Date     | : 24 <sup>th</sup> Feb 2024 |

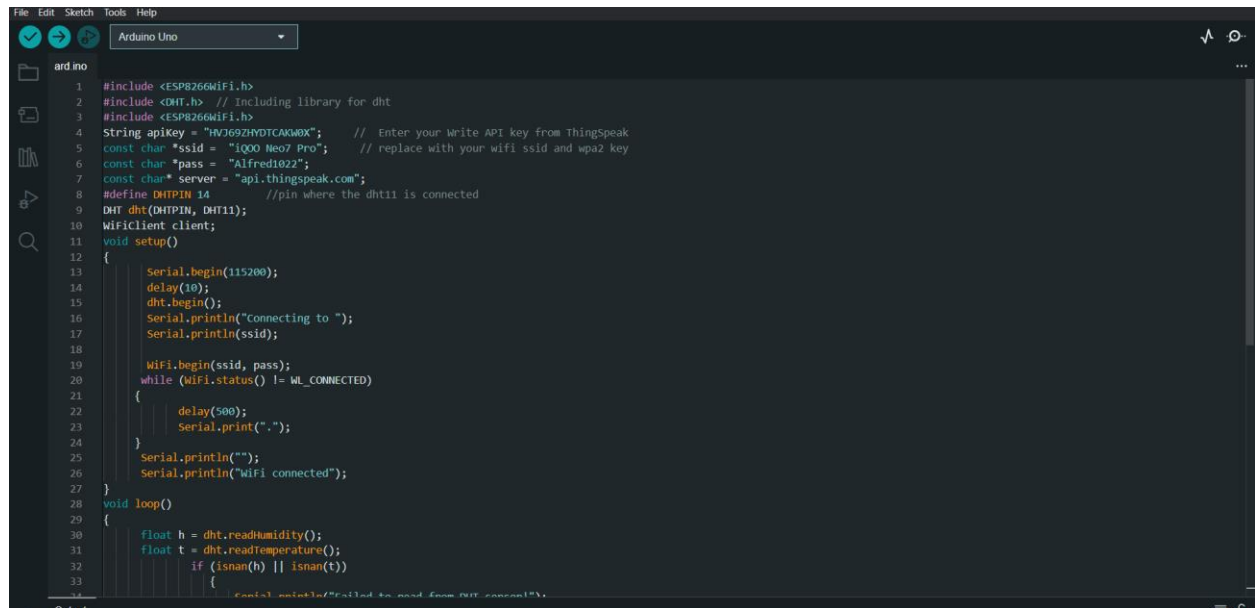
### AIM

Environmental monitoring using NodeMCU with DHT22 and ThingSpeak Cloud Computing.

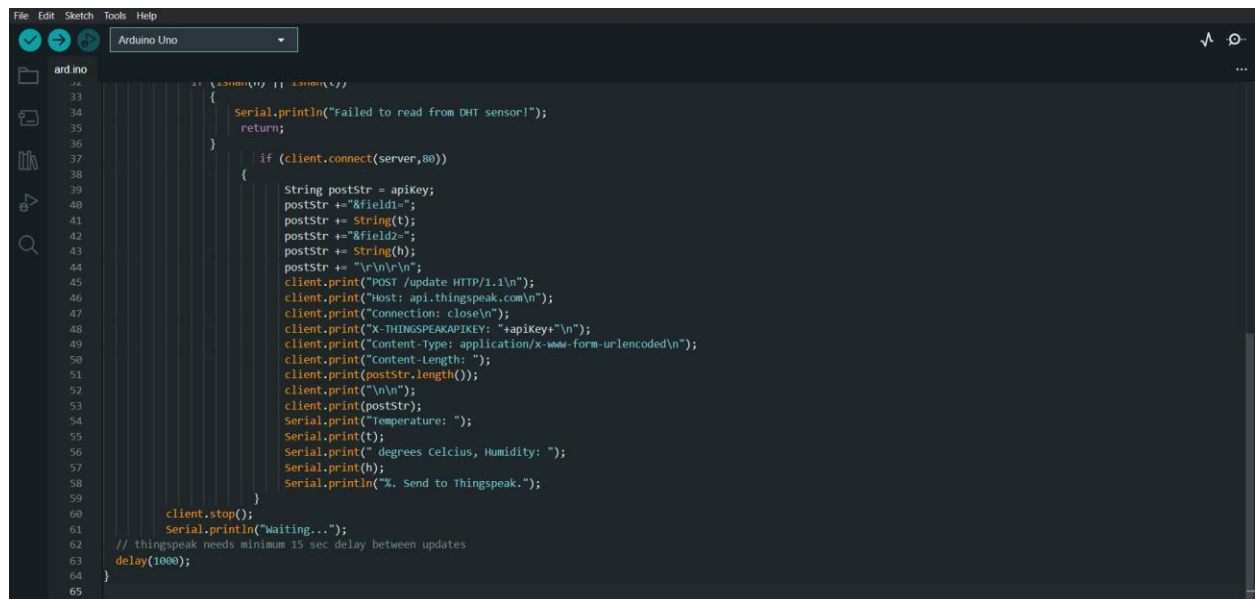
### Steps to Follow

1. Install/update Arduino IDE for NodeMCU compatibility.
2. Install USB-to-Serial bridge driver.
3. Install ESP8266 Arduino Core.
4. Prepare source code by including the required libraries, defining Wifi Credentials and configuring the ThingSpeak API Key:
5. In the Main loop:
  - Read temperature/humidity data.
  - Connect to ThingSpeak server.
  - Send data via HTTP POST.
  - Print data to Serial Monitor.
  - Add 15-second delay.

## Code Screenshots



```
1 #include <ESP8266WiFi.h>
2 #include <DHT.h> // Including library for dht
3 #include <ESP8266WiFi.h>
4 String apiKey = "HVJ69ZHYDTCaKW0X"; // Enter your Write API key from Thingspeak
5 const char *ssid = "iQ00 Neo7 Pro"; // replace with your wifi ssid and wpa2 key
6 const char *pass = "Alfred1022";
7 const char *server = "api.thingspeak.com";
8 #define DHTPIN 14 //pin where the dht11 is connected
9 DHT dht(DHTPIN, DHT11);
10 WiFiClient client;
11 void setup()
12 {
13     Serial.begin(115200);
14     delay(10);
15     dht.begin();
16     Serial.println("Connecting to ");
17     Serial.println(ssid);
18
19     WiFi.begin(ssid, pass);
20     while (WiFi.status() != WL_CONNECTED)
21     {
22         delay(500);
23         Serial.print(".");
24     }
25     Serial.println("");
26     Serial.println("WiFi connected");
27 }
28 void loop()
29 {
30     float h = dht.readHumidity();
31     float t = dht.readTemperature();
32     if (isnan(h) || isnan(t))
33     {
34         Serial.println("Failed to read from DHT sensor!");
35     }
36 }
```



```
37     if (client.connect(server,80))
38     {
39         String postStr = apiKey;
40         postStr += "&field1=";
41         postStr += String(t);
42         postStr += "&field2=";
43         postStr += String(h);
44         postStr += "\n\n";
45         client.print("POST /update HTTP/1.1\n");
46         client.print("Host: api.thingspeak.com\n");
47         client.print("Connection: close\n");
48         client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
49         client.print("Content-Type: application/x-www-form-urlencoded\n");
50         client.print("Content-Length: ");
51         client.print(postStr.length());
52         client.print("\n\n");
53         client.print(postStr);
54         Serial.print("Temperature: ");
55         Serial.print(t);
56         Serial.print(" degrees Celcius, Humidity: ");
57         Serial.print(h);
58         Serial.println("%. Send to Thingspeak.");
59     }
60     client.stop();
61     Serial.println("Waiting...");
62     // thingspeak needs minimum 15 sec delay between updates
63     delay(1000);
64 }
65 }
```

## Source Code

```
#include <ESP8266WiFi.h>
#include <DHT.h> // Including library for dht
#include <ESP8266WiFi.h>
String apiKey = "HVJ69ZHYDTCaKW0X"; // Enter your Write API key from
Thingspeak
const char *ssid = "iQ00 Neo7 Pro"; // replace with your wifi ssid and wpa2
key
```

```

const char *pass = "Alfred1022";
const char* server = "api.thingspeak.com";
#define DHTPIN 14          //pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup()
{
    Serial.begin(115200);
    delay(10);
    dht.begin();
    Serial.println("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        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(t);
        postStr += "&field2=";
        postStr += String(h);
        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");
    }
}

```

```

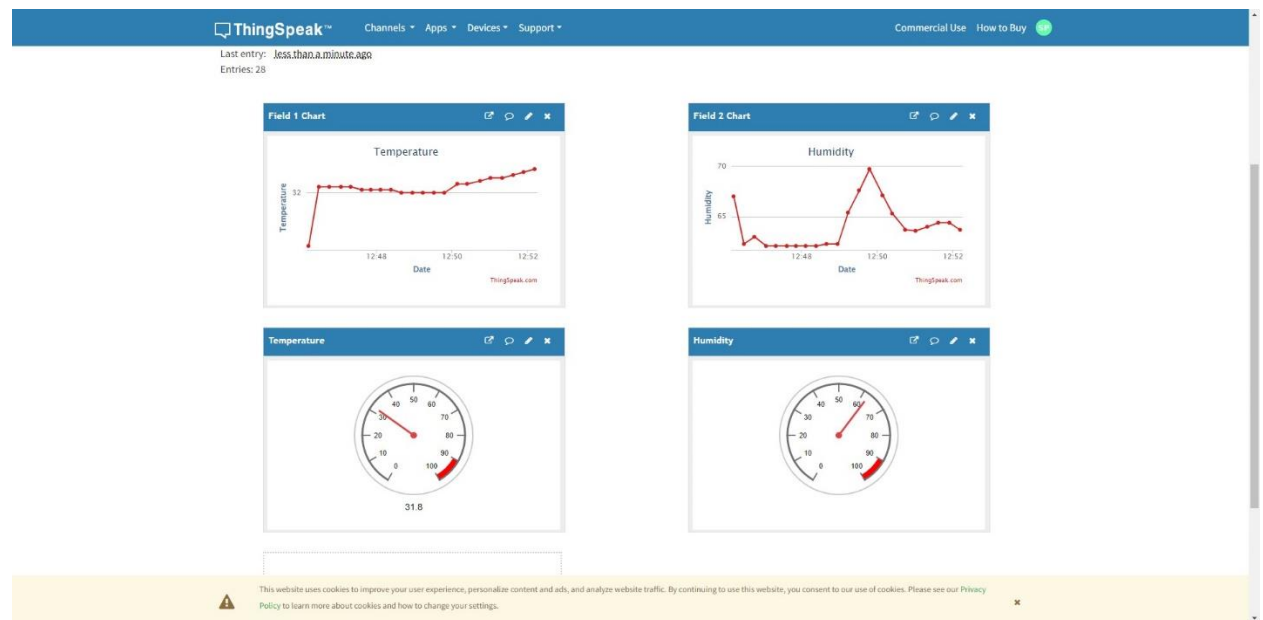
        client.print(postStr);
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.print(" degrees Celcius, Humidity: ");
        Serial.print(h);
        Serial.println("%. Send to Thingspeak.");
    }
    client.stop();
    Serial.println("Waiting...");
    delay(1000);
}

```

## Features and Description

| Feature                     | Description   |
|-----------------------------|---|
| 15 ADC channels             | 15 channels of 12-bit SAR ADC with selectable ranges of 0-1V, 0-1.4V, 0-2V, or 0-4V   |
| 2 UART interfaces           | 2 UART interfaces with flow control and IrDA support  |
| 25 PWM outputs              | 25 PWM pins to control things like motor speed or LED brightness  |
| 2 DAC channels              | Two 8-bit DACs to generate true analog voltages   |
| SPI, I2C, and I2S interface | Three SPI and one I2C interfaces for connecting various sensors and peripherals, as well as two I2S interfaces for adding sound to your project |
| 9 Touch Pads                | 9 GPIOs with capacitive touch sensing   |

## Output Screenshot



## Results

Thus, Environmental monitoring using NodeMCU with DHT22 and ThingSpeak Cloud Computing has been performed and verified.