CHAPTER:17 THINGSPEAK

PRACTICAL: 17A

AIM: Plotting data on Thingspeak.com.

ARDUINO CODE :

```
/**********************
 * Author: Shreejicharan
 * Title:  Plotting data on thingspeak.com: Take analog input from ESP and pass that data to
api.thingspeak.com and prepare an online graph.
 * Date: 27/05/2017
 * Time: 6:00
 * Email: shreejicharanelectronics@gmail.com
 **********************/

/* Plotting data on thingspeak.com: Take analog input from ESP and pass that data
 * to api.thingspeak.com and prepare an online graph.
 *
 */
#include <ESP8266WiFi.h>

#define SENSOR A0

const char* ssid     = "ketan";
const char* password = "dipali@123";

const char* host = "api.thingspeak.com";

const char* privateKey = "06GEN7OPTYMQIJHO";

void setup() {
  Serial.begin(9600);
  delay(10);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

float value = 0;

void loop() {
  delay(5000);
  value = analogRead(A0);

  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  Serial.println("connection done");
  // We now create a URI for the request
  String url = "/update?";
  url += "key=";
  url += privateKey;
  url += "&field1=";
  url += value;

  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(10);
  // Read all the lines of the reply from server and print them to Serial
  while (client.available()) {
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }

  Serial.println();
  Serial.println("closing connection");
}
```

SIMULATION:

CHAPTER:17 THINGSPEAK

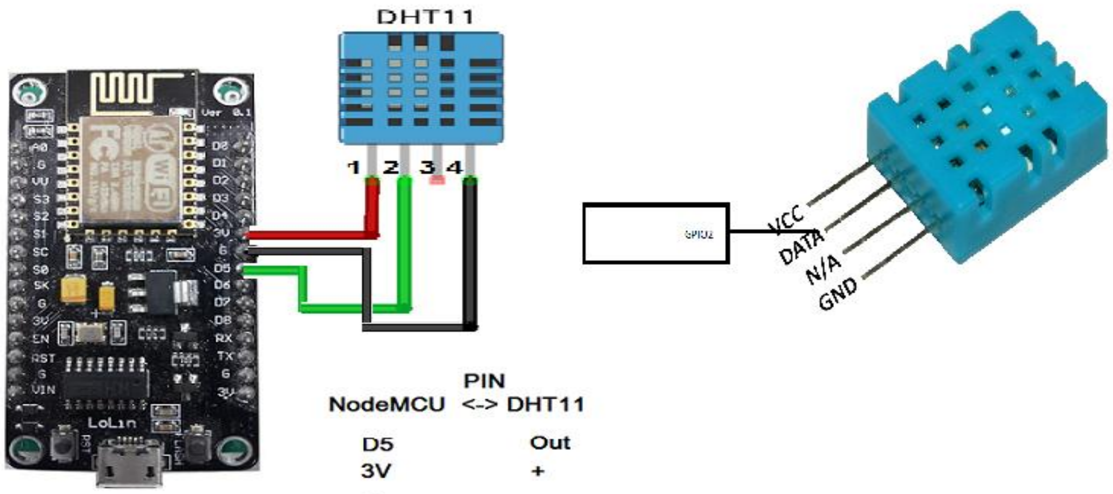
PRACTICAL: 17B

AIM: Plotting DHT11 Sensor data on Thingspeak.com.

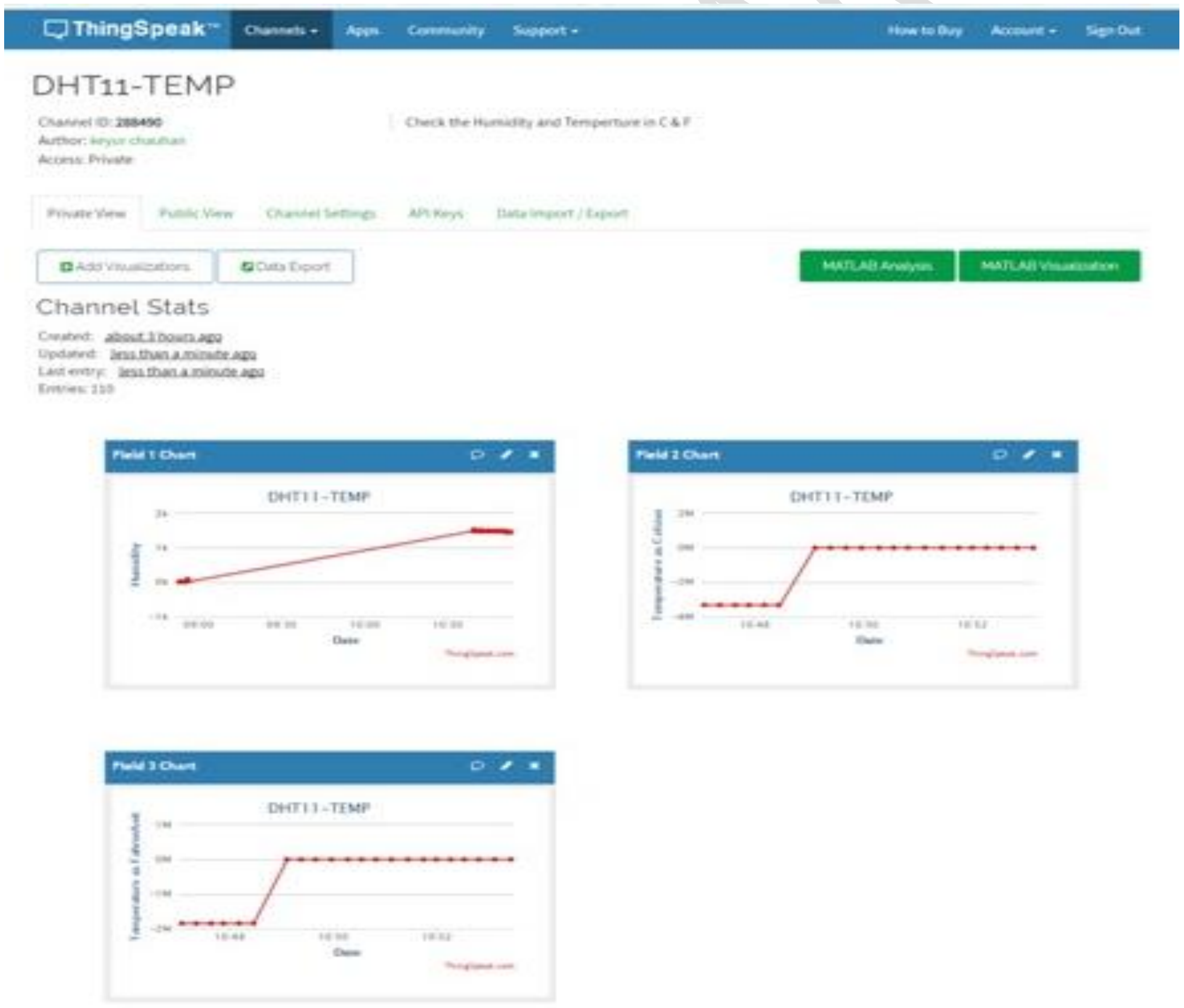
ARDUINO CODE:

```
/**********************
 * Author: Shreejicharan
 * Title:  Plotting data on thingspeak.com: Take analog input from ESP and pass that data to
api.thingspeak.com and prepare an online graph.
 * Date: 27/05/2017
 * Time: 6:00
 * Email: shreejicharanelectronics@gmail.com
 **********************/

/* Plotting data on thingspeak.com: Take analog input from ESP and pass that data
 * to api.thingspeak.com and prepare an online graph.
 *
 */
#include <ESP8266WiFi.h>
#include "DHT.h"

#define DHTPIN 5     // what digital pin we're connected to

#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321

DHT dht(DHTPIN, DHTTYPE);

// Replace with your network details
const char* ssid = "ketan";
const char* password = "dipali@123";

const char* host = "api.thingspeak.com";

const char* privateKey = "CMZHTZ9GRATS8H08";

void setup() {
  Serial.begin(9600);
  delay(10);
//Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

float value = 0;

void loop() {
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hic);
  Serial.print(" *C ");
  Serial.print(hif);
  Serial.println(" *F");

  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  Serial.println("connection done");
  // We now create a URI for the request
  String url = "/update?";
  url += "key=";
  url += privateKey;
  url += "&field1=";
  url += h;
  url += "&field2=";
  url += t;
  url += "&field3=";
  url += f;
  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(10);

  // Read all the lines of the reply from server and print them to Serial
  while (client.available()) {
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }

  Serial.println();
  Serial.println("closing connection");
}
```

SIMULATION:



NodeMCU	PIN <-> DHT11
D5	Out
3V	+
G	-



DHT11-TEMP

Channel ID: 288490
Author: keyur chauhan
Access: Private

Check the Humidity and Tempertuer in C & F

Channel Stats

Created: about 3 hours ago
Updated: less than a minute ago
Last entry: less than a minute ago
Entries: 110