Aim: Upload temperature and humidity data to the cloud using Arduino.

Hardware Requirements:

1. Arduino UNO board

2. NodeMCU ESP8266 Breakout Board 3. DHT-11 temperature and humidity sensor

4. Jumper wires

5. Bread board

6. WIFI Network

Procedure: 1. Download esp8266 Zip file->go to libraries->add Zip file

2. Connect Node MCU, Go to tools->change board to Node MCU esp8266 and port numb

3. Connect DHT-11 temperature and Humidity sensor to Node MCU

4. Sign up to cloud->open THINGSPEAK>create channels->copy API key to the source

5. SSID and password of your WIFI connection should be given in source code

6. Compile and upload the program and verify the temperature and humidity readings in s

monitor 7. Go to cloud and verify the temperature and humidity values in graph.

SOURCE CODE:

Secrets.h

#define SECRET\_SSID "ACENAARTECHNOLOGY" // replace MySSID with you

WiFi network name #define SECRET\_PASS "987654321" // replace MyPassword with your WiFi

password #define SECRET\_CH\_ID 1746705 // replace 0000000 with your channe

number #define SECRET\_WRITE\_APIKEY "NOXKWYKGYZ75KC2S" // replace XYZ with channel write API Key

MAIN CODE:

#include <ESP8266WiFi.h>

#include "secrets.h"

#include "ThingSpeak.h"

#include "DHT.h"

#define DHTTYPE DHT11

#define DHTPIN\_D3

DHT dht(D3, DHT11);

char ssid[] = SECRET\_SSID;

char pass[] = SECRET\_PASS;

int keyIndex = 0;

WiFiClient client;

unsigned long myChannelNumber = SECRET\_CH\_ID;

const char \* myWriteAPIKey = SECRET\_WRITE\_APIKEY;

// Initialize our values

int number1 = 0;

int number2 = random(0,100);

int number3 = random(0,100);

int number4 = random(0,100);

String myStatus = "";

void setup() {

Serial.begin(115200);

while (!Serial) {

;

}

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

}

void loop() {

if(WiFi.status() != WL\_CONNECTED){

Serial.print("Attempting to connect to SSID: ");

Serial.println(SECRET\_SSID);

while(WiFi.status() != WL\_CONNECTED){

WiFi.begin(ssid, pass);

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

}

// set the fields with the values

ThingSpeak.setField(1, number1);

ThingSpeak.setField(2, number2);

ThingSpeak.setField(3, number3);

ThingSpeak.setField(4, number4);

if(number1 > number2){

myStatus = String("field1 is greater than field2");

}

else if(number1 < number2){

myStatus = String("field1 is less than field2");

}

else{

myStatus = String("field1 equals field2");

}

ThingSpeak.setStatus(myStatus);

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

if(x == 200){

Serial.println("Channel update successful.");

}

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

number1++;

if(number1 > 99){

number1 = 0;

}

number2 = random(0,100);

number3 = random(0,100);

number4 = random(0,100);

delay(5000);

}