Expt no 7:write a program to upload temprature and humdity data to the cloud using arduino or raspberry pi.

Procedure:

1. file->preferences ->additional boards manager URLs arduino.esp8266 link is copied.
2. Sketch->include library->manage library->thing speak library install.
3. Tools->board node mcu1.0 ->board manager ->ESP8266 install.
4. File->Examples->Thingspeak->ESP8266->program board directly->write multiple fields.
5. Tools->board node MCU1.0->board manager->node MCU1.0
6. Tools->board node MCU1.0->board manager->Arduino uno is selected.
7. Tools->port is selected.

DHT 11 sensor pin connection:

+ - vin

Out- D3

- - GND

Write multiple fields:

#include <ESP8266WiFi.h>

#include "secrets.h"

#include "ThingSpeak.h"

#include "DHT.h"

#define DHTTYPE DHT11

#define DHTPIN\_ D3

DHT dht(D3,DHT11);

// always include thingspeak header file after other header files and custom macros

char ssid[] = SECRET\_SSID; // your network SSID (name)

char pass[] = SECRET\_PASS; // your network password

int keyIndex = 0;

DHT dht(DHTPIN, DHTTYPE);// your network key Index number (needed only for WEP)

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() {

dht.begin();

Serial.begin(115200); // Initialize serial

while (!Serial) {

; // wait for serial port to connect. Needed for Leonardo native USB port only

}

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client); // Initialize ThingSpeak

}

void loop() {

// Connect or reconnect to WiFi

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); // Connect to WPA/WPA2 network. Change this line if using open or WEP network

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);

// figure out the status message

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");

}

// set the status

ThingSpeak.setStatus(myStatus);

// write to the ThingSpeak channel

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));

}

// change the values

number1++;

if(number1 > 99){

number1 = 0;

}

number2 = random(0,100);

number3 = random(0,100);

number4 = random(0,100);

delay(20000); // Wait 20 seconds to update the channel again

}

Secret.h

// Use this file to store all of the private credentials

// and connection details

#define SECRET\_SSID "samsung on7 pro" // replace MySSID with your WiFi network name

#define SECRET\_PASS "Venkata123#" // replace MyPassword with your WiFi password

#define SECRET\_CH\_ID 2040479 // replace 0000000 with your channel number

#define SECRET\_WRITE\_APIKEY "0RNY24517GJ9LRWD" // replace XYZ with your channel write API Key