Experiment No: 5

Node MCU /Arduino / Raspberry Pi to upload the temperature data on cloud.

Name of the Student: \_Manthan Khawse \_ Div. CS-C & Roll No. 03

**Aim**: Node MCU /Arduino / Raspberry Pi to upload the temperature data on cloud.

# Components Required:

1. Node MCU – 1
2. Micro USB Cable – 1
3. PC/Laptop – 1
4. Connecting Wires
5. Bread Board – 1
6. Temperature Sensor LM 35

# Software Required:

Arduino IDE

**Theory:** We all are observed LM 35 temperature sensor during practical no. 3.

# Procedure:

Step 1: Include Wi-Fi and ThingSpeak directories.

Step 2: Note that we already have wi-fi directory installed. If not, install it.

Step 3: Installing Thingspeak. Goto Sketch---Include Library---Manage Libraries---Write in Library Manage ‘Thingspeak’---Install latest version

Step 4: Goto [www.google.com.](http://www.google.com/) Googlr -thingspeak login---Sign-in

Step 5: Create account if you don’t have one. Use vit.edu mailID. Location India. Your name etc--- Continue

Step 6: Goto Channels --- My Channels--- New Channel

Step 7: Write Channel name, description (not mandatory)---Create two fields. Field 1- temp in degrees Celsius. Field 2- temp in Fahrenheit. Save.

Step 8: Copy Channel ID and paste it in the code. long myChannelNumber = 2378063;

Step 9: Goto API keys. Copy API key (Write API Key) and paste it in the code. const char myWriteAPIKey[] = "PFZ2AGFH7FJFROBO";

Step 10: Enter the wifi login and password in the code. (Same as in Expt 4). WiFi.begin("Login","Password");

Step 11: Write code for reading data from LM35 temperature sensor (Same as in Expt 3). ThingSpeak.begin(client); ------ Starts thingspeak ThingSpeak.writeField (myChannelNumber, 1, tempc, myWriteAPIKey); Displays temp in the field in thingspeak.

Step 12: Make hardware connections using node MCU and LM35, to sense and measure temperature.

Step 13: Upload sketch. The data (temp) will be displayed in the serial monitor. Also it will be collected and uploaded on cloud and displayed in the two fields.

Step 13: Observe the outputs.

1] Using LM35 sensor:

#include <ESP8266WiFi.h> #include <ThingSpeak.h>

const char\* ssid = "Galaxy M13 5G 3CEC";

const char\* password = "manthan@123";

const char myWriteAPIKey[] ="PFZ2AGFH7FJFROBO"; unsigned long myChannelNumber = 2814271; WiFiClient client;

#define LM35\_PIN A0 void setup() {

Serial.begin(9600);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) { delay(1000);

}

ThingSpeak.begin(client);

}

void loop() {

int analogValue = analogRead(LM35\_PIN);

float millivoltage = analogValue \* (3.3 / 1024.0) \* 3300; float tempC = millivoltage / 10.0;

float tempF = (tempC \* 9 / 5) + 32;

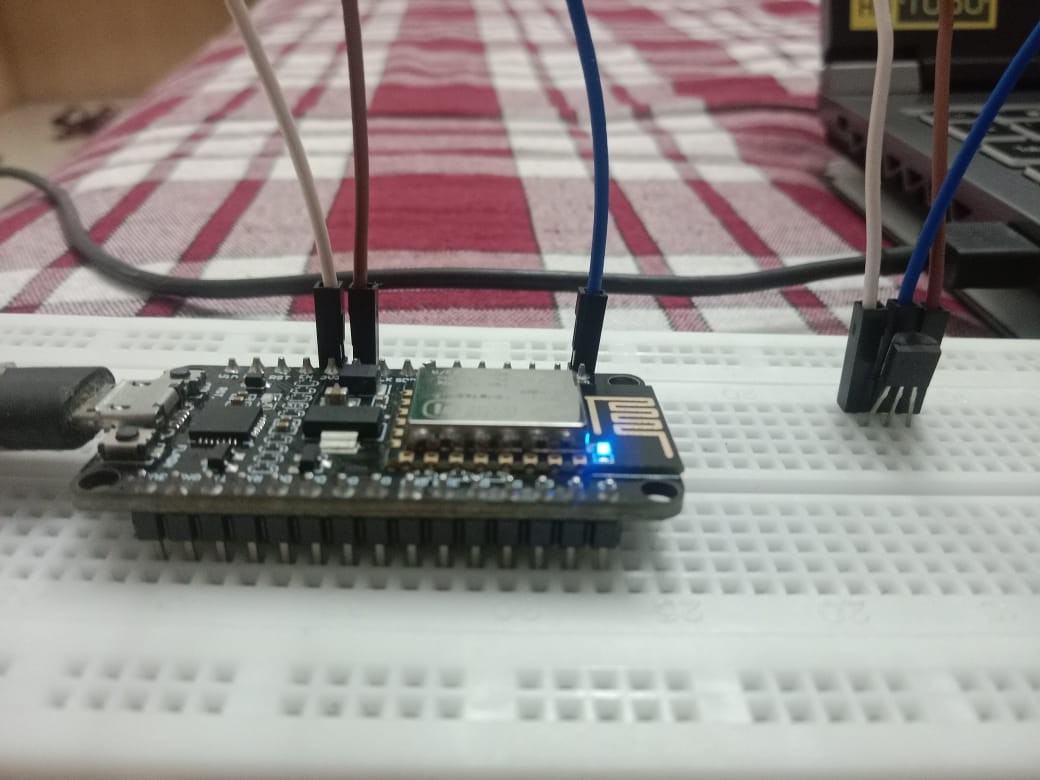
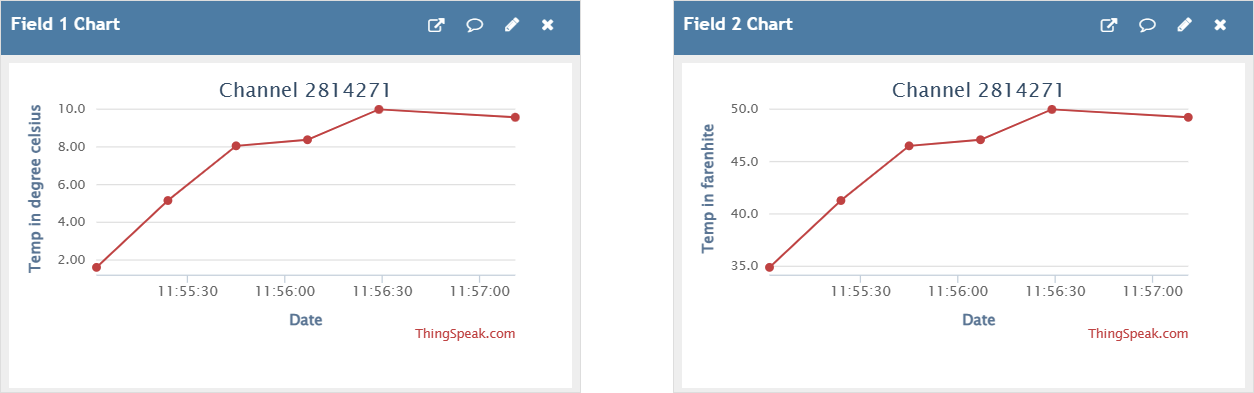
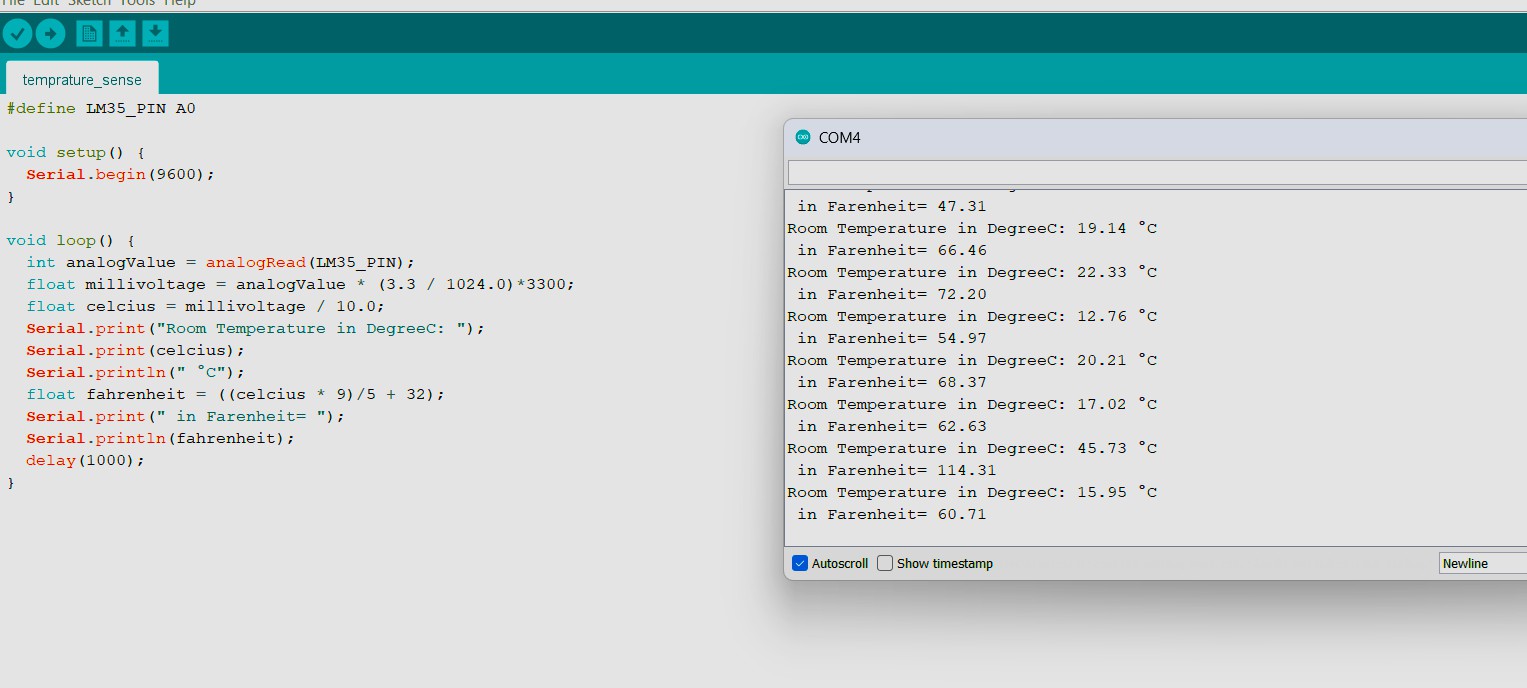
ThingSpeak.setField(1, tempC); ThingSpeak.setField(2, tempF);

ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

Serial.print("Temperature (°C): "); Serial.println(tempC); Serial.print("Temperature (°F): "); Serial.println(tempF);

delay(20000);

}



**Conclusion:**

**This code reads an LM35 sensor value using a NodeMCU ESP8266 and uploads the data to ThingSpeak for real-time monitoring. It continuously captures and transmits the sensor readings over WiFi, making it suitable for IoT applications.**