|  |  |
| --- | --- |
| Name | Shaunak Deshpande |
| Division | IC-C |
| Roll number | 39 |
| Batch | 2 |
| GR Number | 11911180 |

**Lab Assignment No.5**

**Title: Sensor LM35 interfacing with ESP32 and sends this data to the Thingspeak.**

**Introduction:**

**LM35 Temperature Sensor :**

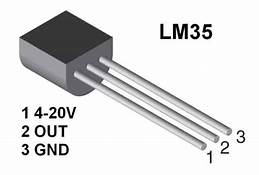
The LM35 is a temperature sensor widely used in electronic projects and midrange devices. It has limited usage in industrial applications due to maximum temperature range limitations. It’s rated to a full range of −55°C to 150°C.

You can just power it up and instantly read the voltage level on the output terminal. The VOUT of the sensor directly maps to the sensor’s temperature.

**Technical Features of LM35 Temperature Sensor :**

* Linear + 10-mV/°C Scale Factor
* 0.5°C Ensured Accuracy (at 25°C)
* Rated for Full −55°C to 150°C Range
* Operates From 4 V to 30 V
* Less Than 60-μA Current Drain
* Non-Linearity Only ±¼°C Typical

#### LM35 Pinout:



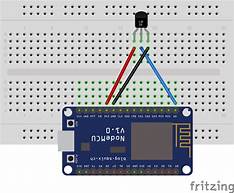
## ESP32 LM35 Temperature Sensor Interfacing (Arduino):

#### ESP32 LM35 Temperature Sensor Connection

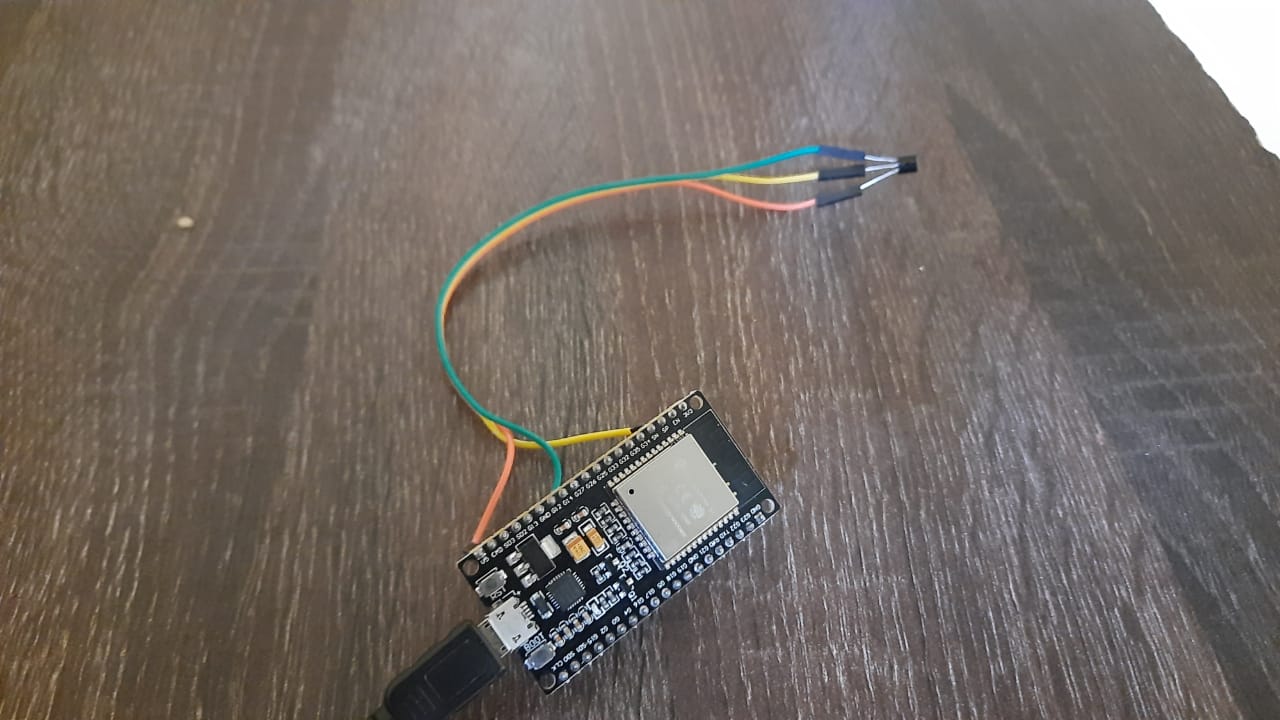
The connection between ESP32 & LM35 temperature sensor should be as follows.

|  |  |
| --- | --- |
| **LM35 Temperature Sensor** | **ESP32 DevKit v1 Board** |
| **VOUT** | GPIO35 |
| **VCC** | Vin |
| **GND** | GND |

**Circuit diagram :**



**Actual Hardware :**



**Arduino Code :**

#include <WiFi.h>

#define ADC\_VREF\_mV 3300.0 // in millivolt

#define ADC\_RESOLUTION 4096.0

#define PIN\_LM35 36

#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros

#define SECRET\_SSID "Galaxy M21A398" // replace MySSID with your WiFi network name

#define SECRET\_PASS "55555555" // replace MyPassword with your WiFi password

#define SECRET\_CH\_ID 1603613 // replace 0000000 with your channel number

#define SECRET\_WRITE\_APIKEY "1EJRT7KAKIR8YZ1M" // replace XYZ with your channel write API Key

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

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

int keyIndex = 0; // your network key Index number (needed only for WEP)

WiFiClient client;

unsigned long myChannelNumber = SECRET\_CH\_ID;

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

void setup() {

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.");

}

int adcVal = analogRead(PIN\_LM35);

// convert the ADC value to voltage in millivolt

float milliVolt = adcVal \* (ADC\_VREF\_mV / ADC\_RESOLUTION);

// convert the voltage to the temperature in °C

float tempC = milliVolt / 10;

// convert the °C to °F

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

// print the temperature in the Serial Monitor:

Serial.print("Temperature: ");

Serial.print(tempC); // print the temperature in °C

Serial.print("°C");

Serial.print(" ~ "); // separator between °C and °F

Serial.print(tempF); // print the temperature in °F

Serial.println("°F");

delay(500);

int x = ThingSpeak.writeField(myChannelNumber, 1, tempC, myWriteAPIKey);

if(x == 200){

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

}

else{

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

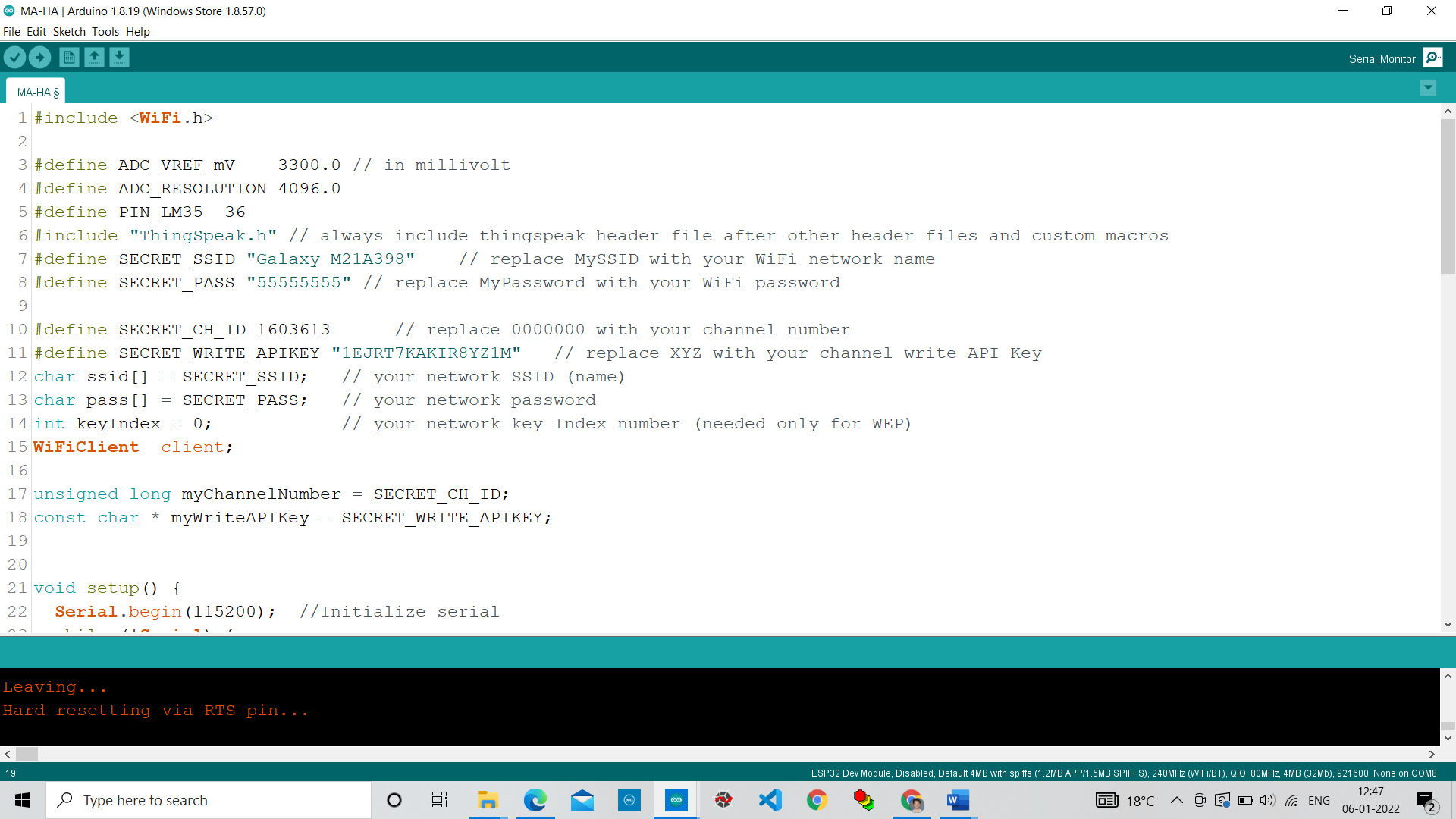
}

// change the value

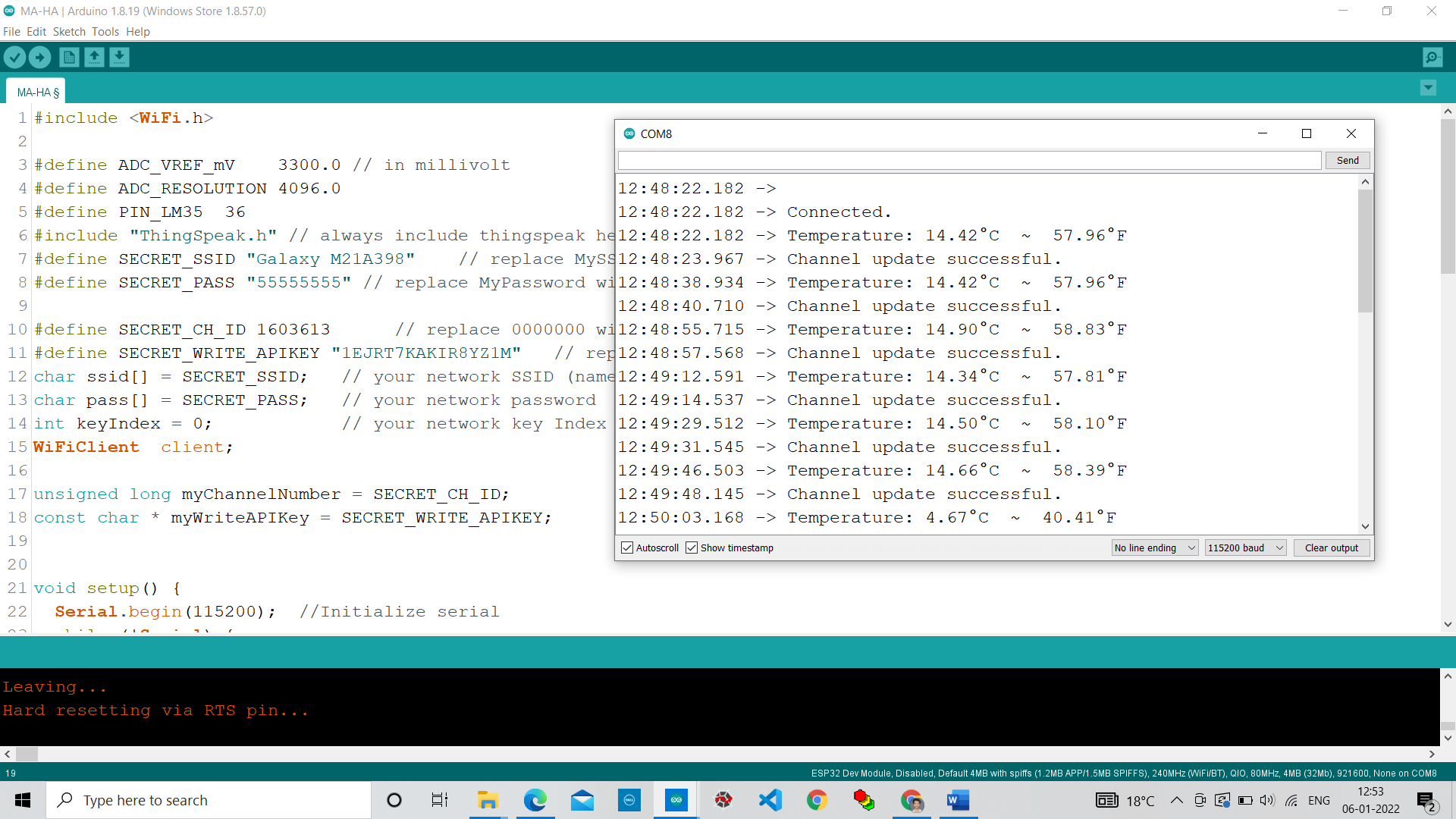
delay(15000); // Wait 20 seconds to update the channel again

}

**Actual Code in Arduino IDE:**

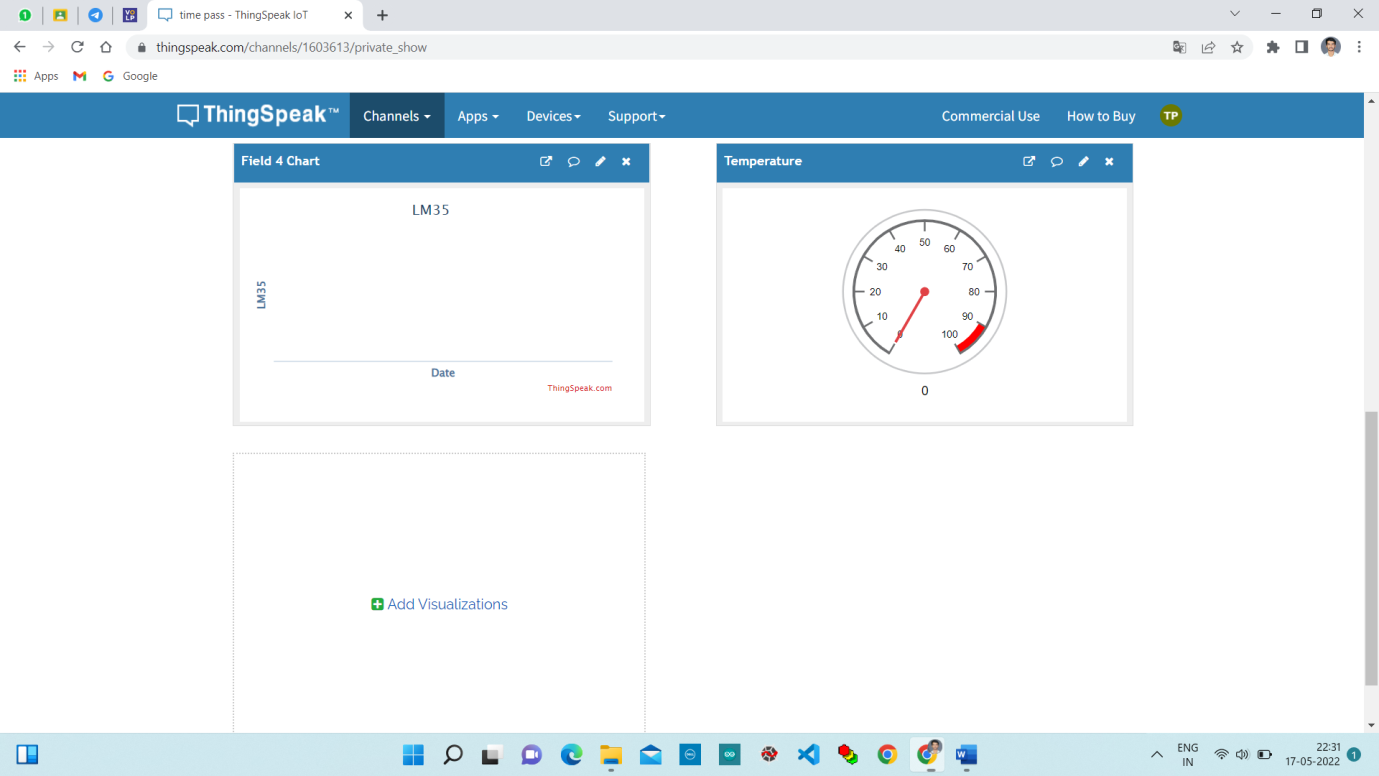


**Temperature Readings at serial monitor:**



**Data send to the Thingspeak:**

In this temperature readings from LM35 is transferred through ESP32 to the thindspeak. According to the variations in temperature graph will change automatically.



**Conclusion:**

In this lab we studied about sensor interfacing with ESP32 and sends this signal to thingseak through cloud and observe the readings on it.