# EPS32 Temperature Detection IoT ThingSpeak 2

## Wiring

A circuit board with wires

Description automatically generated

## Code

//channel ID: 2587382

// channel api key: 4V3J6GQGEJ7JWFE5

#include <WiFi.h>           // Include the WiFi library

#include <ThingSpeak.h>     // Include the ThingSpeak library for sending data to ThingSpeak

#include "DHTesp.h"         // Include the DHT sensor library

#define DHTPIN 15           // Define the pin where the DHT sensor is connected

#define LED\_PIN 2           // Define the pin where the LED is connected

DHTesp dht;                 // Create a DHTesp object

const char\* ssid = "Wokwi-GUEST";  // Define the WiFi SSID

const char\* password = "";         // Define the WiFi password (empty for Wokwi-GUEST)

// ThingSpeak settings

unsigned long myChannelNumber = 2587382;  // Define the ThingSpeak channel number

const char\* myWriteAPIKey = "4V3J6GQGEJ7JWFE5";  // Define the ThingSpeak Write API Key

WiFiClient  client;         // Create a WiFiClient object

void setup() {

  pinMode(LED\_PIN, OUTPUT);  // Set the LED pin as an output

  digitalWrite(LED\_PIN, LOW); // Turn off the LED initially

**Serial**.begin(9600);        // Initialize serial communication at 9600 baud rate

  dht.setup(DHTPIN, DHTesp::DHT22);  // Initialize the DHT sensor

  WiFi.begin(ssid, password);  // Connect to WiFi network

**Serial**.print("Connecting to WiFi");

  while (WiFi.status() != WL\_CONNECTED) {  // Wait for the connection to establish

    delay(1000);

**Serial**.print(".");

  }

**Serial**.println();

**Serial**.println("Connected to WiFi");

  ThingSpeak.begin(client);  // Initialize ThingSpeak

}

void loop() {

  float temperature = dht.getTemperature();  // Read temperature from the DHT sensor

  if (isnan(temperature)) {                   // Check if the reading is valid

**Serial**.println("Failed to read from DHT sensor!");  // Print error message if reading fails

    temperature = 0.0;                                   // Set temperature to 0.0 if reading fails

  }

**Serial**.print("Temperature: ");  // Print the temperature value

**Serial**.println(temperature);

  if (temperature > 30) {            // Check if the temperature is greater than 30°C

    digitalWrite(LED\_PIN, HIGH);     // Turn on the LED if temperature is above 30°C

  } else {

    digitalWrite(LED\_PIN, LOW);      // Turn off the LED if temperature is 30°C or below

  }

  // Send the temperature value to ThingSpeak

  ThingSpeak.setField(1, temperature);

  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));

  }

  delay(20000);  // Wait for 20 seconds before the next update (ThingSpeak allows updates every 15 seconds)

}

## Library

# Wokwi Library List

# See https://docs.wokwi.com/guides/libraries

# Automatically added based on includes:

**WiFi**

**DHT sensor library for ESPx**

**ThingSpeak**

ThingSpeak

A screenshot of a computer

Description automatically generated

A screenshot of a channel settings

Description automatically generated

A screenshot of a computer

Description automatically generated