#include <WiFi.h>

#include <HTTPClient.h>

// WiFi Credentials

const char\* ssid = "YOUR\_WIFI\_SSID";

const char\* password = "YOUR\_WIFI\_PASSWORD";

// ThingSpeak API Details

const char\* server = "https://api.thingspeak.com/update?api\_key=YOUR\_THINGSPEAK\_WRITE\_API\_KEY";

// Sensor Pins

#define TRIG\_PIN 14 // GPIO14

#define ECHO\_PIN 12 // GPIO12

#define GAS\_SENSOR 34 // GPIO34 (Analog input for MQ-135)

long duration;

float distance;

int gasValue;

void setup() {

Serial.begin(115200);

pinMode(TRIG\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi");

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

delay(1000);

Serial.print(".");

}

Serial.println("\nConnected to WiFi!");

}

void loop() {

// Measure distance with Ultrasonic Sensor

digitalWrite(TRIG\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG\_PIN, LOW);

duration = pulseIn(ECHO\_PIN, HIGH);

distance = duration \* 0.034 / 2;

// Read gas sensor value

gasValue = analogRead(GAS\_SENSOR);

Serial.print("Garbage Level (cm): ");

Serial.println(distance);

Serial.print("Gas Sensor Value: ");

Serial.println(gasValue);

// Send data to ThingSpeak

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

HTTPClient http;

String url = String(server) + "&field1=" + String(distance) + "&field2=" + String(gasValue);

http.begin(url);

int httpResponseCode = http.GET();

if (httpResponseCode > 0) {

Serial.println("Data sent to ThingSpeak!");

} else {

Serial.print("Error sending data: ");

Serial.println(httpResponseCode);

}

http.end();

} else {

Serial.println("WiFi Disconnected");

}

delay(15000); // ThingSpeak allows data updates every 15 seconds

}