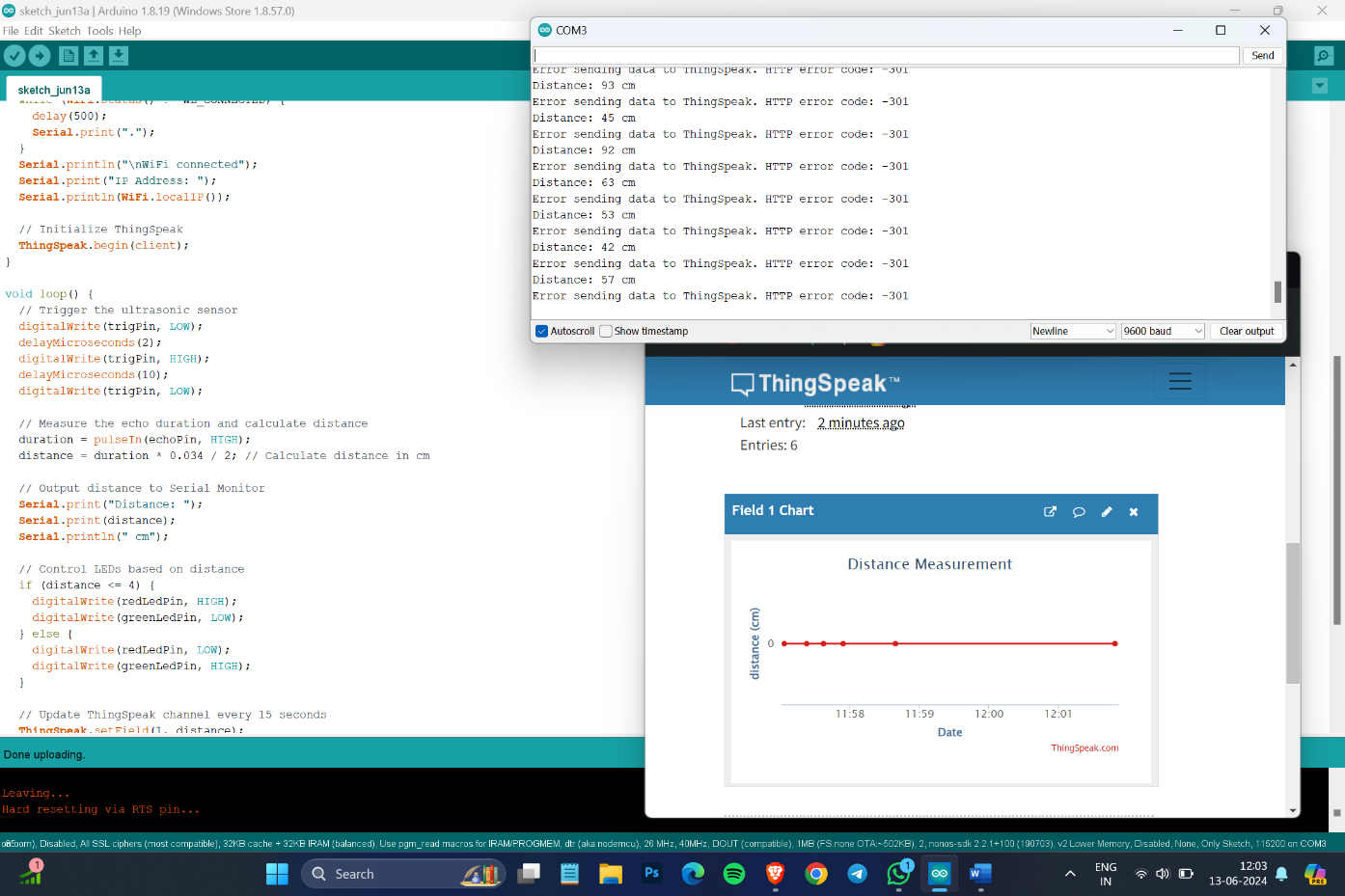
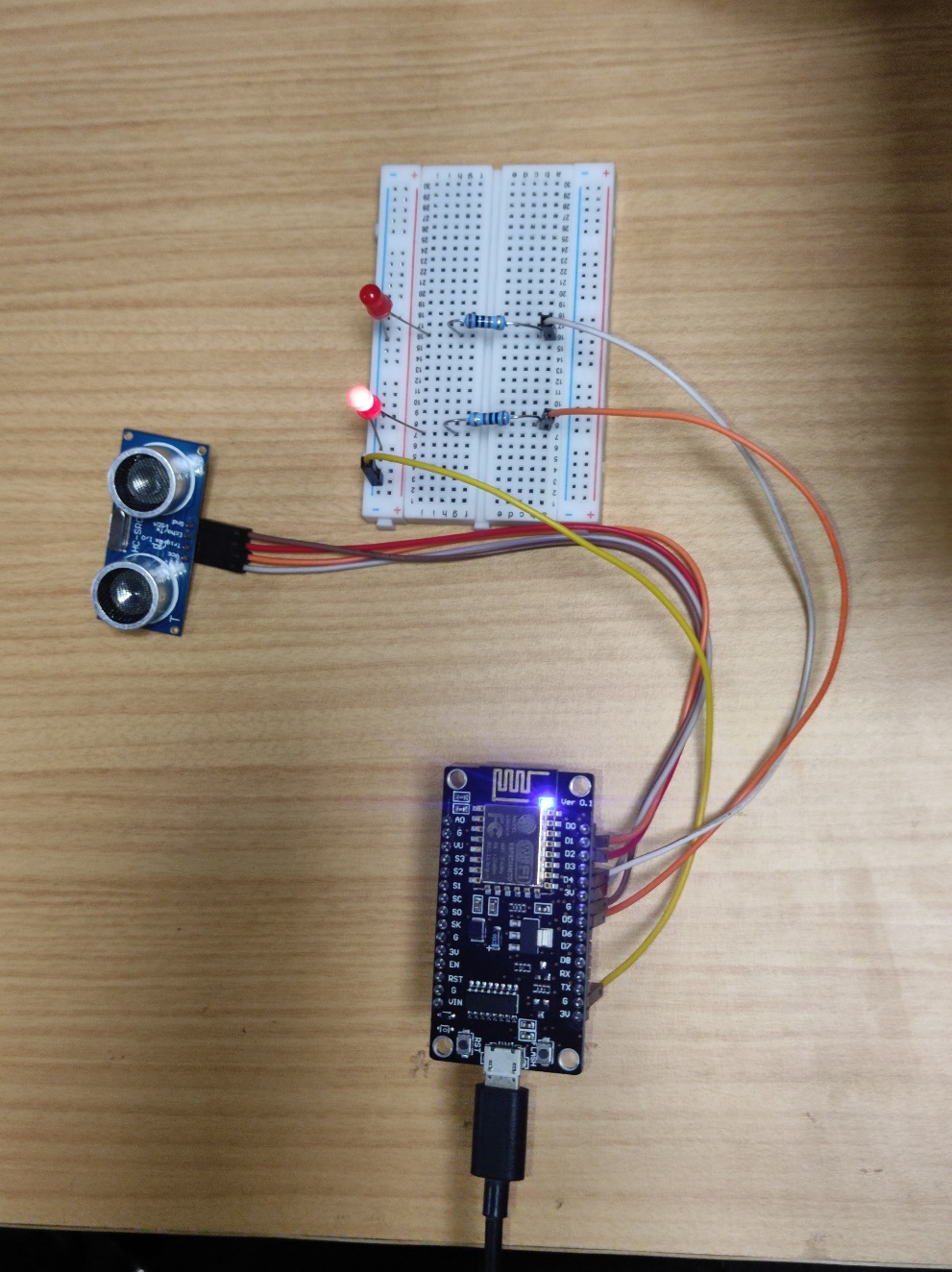
**ESP32 – Powered Flood Monitoring System**





**Powered Flood Monitoring System code** #include <ESP8266WiFi.h>

#include <ThingSpeak.h>

const int trigPin = 5; // Ultrasonic sensor trigger pin (D1)

const int echoPin = 4; // Ultrasonic sensor echo pin (D2)

const int redLedPin = 2; // Red LED pin (D4)

const int greenLedPin = 14; // Green LED pin (D5)

unsigned long channelNumber =2576388; // ThingSpeak channel number

const char \*writeAPIKey = "160DA9PNZ81RNUMJ"; // ThingSpeak write API key

const char\* ssid = "Pavan X2";

const char\* password = "123456788";

WiFiClient client;

long duration;

int distance;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(redLedPin, OUTPUT);

pinMode(greenLedPin, OUTPUT);

digitalWrite(redLedPin, LOW);

digitalWrite(greenLedPin, LOW);

Serial.begin(9600);

// Attempt to connect to WiFi

WiFi.begin(ssid, password);

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

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

delay(500);

Serial.print(".");

}

Serial.println("\nWiFi connected");

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

// Initialize ThingSpeak

ThingSpeak.begin(client);

}

void loop() {

// Trigger the ultrasonic sensor

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Measure the echo duration and calculate distance

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2; // Calculate distance in cm

// Output distance to Serial Monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Control LEDs based on distance

if (distance <= 4) {

digitalWrite(redLedPin, HIGH);

digitalWrite(greenLedPin, LOW);

} else {

digitalWrite(redLedPin, LOW);

digitalWrite(greenLedPin, HIGH);

}

// Update ThingSpeak channel every 15 seconds

ThingSpeak.setField(1, distance);

int response = ThingSpeak.writeFields(channelNumber, writeAPIKey);

// Check if update was successful

if (response == 200) {

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

} else {

Serial.print("Error sending data to ThingSpeak. HTTP error code: ");

Serial.println(response);

}

delay(5000); // Delay 15 seconds before next update

}