Smart water fountain

Code:

#include <WiFi.h>

#include <ThingSpeak.h>

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

WiFiClient client;

unsigned long myChannelNumber =  2326385;

const char \* myWriteAPIKey = "QNCY8FUD8V0NJSG3";

const int pirSensorPin = 23;         // Pin connected to PIR motion sensor

const int ultrasonicTriggerPin = 33;  // Pin connected to the Ultrasonic sensor trigger

const int ultrasonicEchoPin = 12;     // Pin connected to the Ultrasonic sensor echo

const int switchRelayPin = 26;

int pirState = LOW;

int val = 0;

int distance = 0;  // Initialize distance

int StatusCode;

// Variables

bool isMotionDetected = false;  // Flag to track motion detection

void setup() {

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

  pinMode(pirSensorPin, INPUT);

  pinMode(ultrasonicTriggerPin, OUTPUT);

  pinMode(ultrasonicEchoPin, INPUT);

  pinMode(switchRelayPin, OUTPUT);

  digitalWrite(switchRelayPin, LOW);  // Turn off the switch initially

**Serial**.begin(115200);  // Initialize serial communication

}

void loop() {

  connectToCloud();

computeData();

writeData();

delay(1000);

}

void connectToCloud(){

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

**Serial**.print("Attempting to connect");

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

WiFi.begin(ssid, pass);

for(int i=0;i<5;i++) {

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

delay(1000);

}

}

**Serial**.println("\nConnected.");

}

}

void computeData(){

  // Check PIR motion sensor

  val = digitalRead(pirSensorPin);  // Read input value

  if (val == HIGH) {  // Check if the input is HIGH

    digitalWrite(switchRelayPin, HIGH);  // Turn switch ON

    if (pirState == LOW) {

      // We have just turned on

**Serial**.println("Motion detected!");

      // We only want to print on the output change, not state

      pirState = HIGH;

    }

  } else {

    digitalWrite(switchRelayPin, LOW);  // Turn switch OFF

    if (pirState == HIGH) {

      // We have just turned off

**Serial**.println("Motion ended!");

      // We only want to print on the output change, not state

      pirState = LOW;

    }

  }

  // Check Ultrasonic sensor

  distance = measureDistance();

}

int measureDistance() {

  // Send a pulse to the Ultrasonic sensor

  digitalWrite(ultrasonicTriggerPin, LOW);

  delayMicroseconds(2);

  digitalWrite(ultrasonicTriggerPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(ultrasonicTriggerPin, LOW);

  // Measure the duration of the pulse

  long duration = pulseIn(ultrasonicEchoPin, HIGH);

  // Calculate the distance based on the speed of sound (343 m/s or 0.0343 cm/µs)

  int distance = duration \* 0.0343 / 2;

   if (isMotionDetected || distance <= 100) {

    digitalWrite(switchRelayPin, LOW);  // Turn on the switch

  } else {

    digitalWrite(switchRelayPin, HIGH);   // Turn off the switch

  }

  return distance;

}

 void writeData(){

ThingSpeak.setField(1, pirState);

ThingSpeak.setField(2, distance);

StatusCode = ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);

if(StatusCode == 200) //successful writing code

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

else

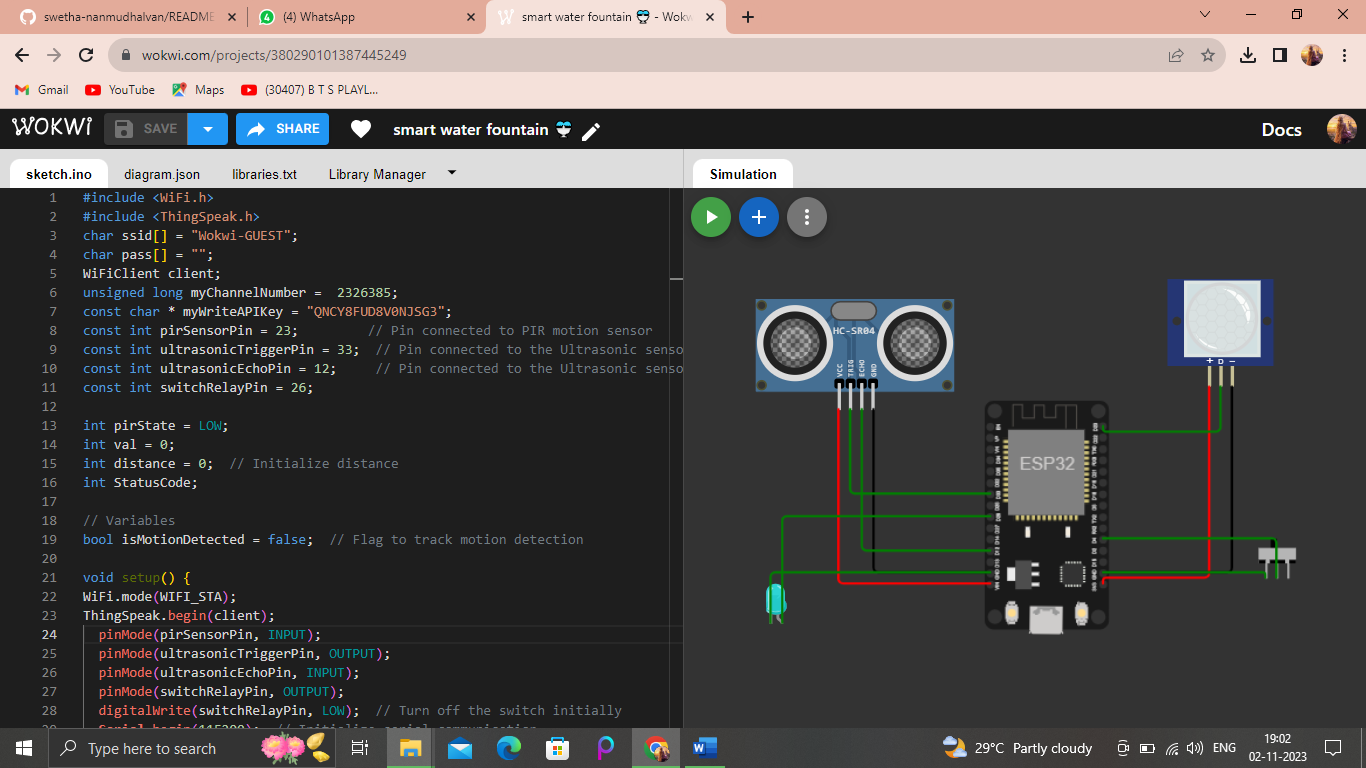
**Serial**.println("Problem Writing data. HTTP error code :" +

String(StatusCode));

delay(15000); // data to be uploaded every 15secs

 }

Stimulation:



Thingspeak:

