****

**Área:** Desarrollo de Software Multiplataforma

**Asignatura:** Aplicaciones de loT

**Docente:** Anastacio Rodríguez García

**Alumnos: Nº Control:**

Sandra Michel Hernández Chiquito 1221100284

Daniela Esmeralda Méndez 1221100344

Omar Alejandro Lozada Zúñiga 1221100256

**Grupo:** GDS0453

**21/04/2023**

**Dolores Hidalgo Cuna de la Independencia Nacional Guanajuato**

**Integrantes:**

Daniela Esmeralda Méndez | 1221100344

Sandra Michel Hernández Chiquito | 1221100284

Omar Alejandro Lozada Zúñiga | 1221100256

**Objetivo general**

El objetivo de este proyecto será tener una casa más segura, ya que, en ocasiones, en la mayoría de los hogares se olvida cerrar el gas de las estufas, lo cual puede ser muy peligro. Por otra parte, también se detectará movimiento y así se iluminará el exterior, de esta manera se pondrá en aviso a quienes están en el interior y harán, por el temor al verse descubierto, que el mal hechor se escape, algo más del proyecto es que cubra una necesidad en tiempos calurosos, detectara la temperatura y humedad y conforme esta sea una temperatura elevada se encenderá un ventilador.

**Objetivos específicos**

* Detectara movimiento al pasar y se encenderán los leds.
* Al detectar temperatura elevada se encenderá un servomotor (para la ventilación).
* El buzzer se encenderá cuando detecte gas o humo.

**Tabla de software utilizado**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Software | Versión | Tipo |
| 001 | Visual Studio Code | 1.71.2 | Freeware |
| 002 | Arduino IDE | 2 | 2 |
| 003 | Librería para sensor temperatura (DHT11) |  |  |
| 004 | Mosquito | 5.0.3.1.1 | Freeware |
| 005 | NODE-RED | 3.0 | Freeware |
| 006 | Motor de base de Datos |  | Freeware |

Tabla de hadware utilizado

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Componentes | Descripción | Imagen | Cantidad | Costo |
| 001 | Esp32 | El módulo ESP32 es una solución de Wi-Fi/Bluetooth todo en uno, integrada y certificada que proporciona no solo la radio inalámbrica, sino también un procesador integrado con interfaces para conectarse con varios periféricos |  | 1 | $160 |
| 002 | Servomotor | Es un tipo de motor eléctrico que tiene la capacidad de controlar el movimiento de su eje, en velocidad, aceleración, par y posición. |  | 1 | $80 |
| 003 | Temperatura  DHT11 | Dispositivos utilizados en aplicaciones de edificación para medir la temperatura de un fluido, normalmente aire o agua. |  | 1 | $75 |
| 004 | Buzzer | Es un pequeño transductor capaz de convertir la energía eléctrica en sonido. |  | 1 | $60 |
| 005 | Led RGB | El Led RGB emite luz a través del paso de la corriente eléctrica, pero tiene la peculiaridad de estar compuesto por tres Leds de diferentes colores: rojo(Red),verde(Green),azul(Blue), de ahí el nombre “RGB” |  | 8 | $24 |
| 006 | MQ-2 | Es un sensor de gas inflamable y humo que detecta las concentraciones de gas combustible en el aire y emite su lectura como un voltaje analógico. |  | 1 | $80 |
| 007 | Movimiento | Es un dispositivo electrónico que pone en funcionamiento un sistema (encendido o apagado) cuando detecta movimiento en el área o ambiente en el que está instalado. |  | 1 | $55 |
| 008 | Small Sound | Es un módulo que detecta sonido y su intensidad. Las aplicaciones de este sensor varían, entre ellas de seguridad, control y monitoreo. |  | 1 | $60 |

**Épicas de historias de Usuario**

* En movimiento que se enciendan las luces para una mejor seguridad.
* También que se detecte tanto el humo como el gas y así sea una señal de alerta
* La casa se ventile dependiendo de una cierta temperatura.

**Tabla Historias de Usuario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Historia de Usuario | Prioridad | Estimación | Como probarlo | Responsable |
| 001 | Al detectar movimiento se enciendan los led | Alta | 1 semana | Al pasar una persona | Esmeralda |
| 002 | La casa se ventile dependiendo de una cierta temperatura. | Media | 2 días | Depende de la temperatura. | Omar |
| 003 | Que la casa detecte fugas de gas haga sonar un buzzer cuando el nivel sea alto | Alta | 3 días | Al exponer el sensor a una fuga de gas. | Michel |

## **Prototipo en dibujo**



**Código**

#include <Servo.h>

#include <DHT.h>

#include <WiFi.h>

#include <PubSubClient.h>

#include <HTTPClient.h>

#include <UrlEncode.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#define DHTPIN 13        // Pin de datos del sensor DHT11

#define DHTTYPE DHT11    // Tipo de sensor DHT

#define SERVOPIN 14      // Pin de señal del servo

#define SERVOMIN 0       // Valor mínimo para el servo

#define SERVOMAX 180     // Valor máximo para el servo

#define TEMP\_MIN 1       // Temperatura mínima para encender el ventilador

#define TEMP\_MAX 20      // Temperatura máxima para apagar el ventilador

#define HUM\_MIN 40       // Humedad mínima para encender el ventilador

#define HUM\_MAX 60       // Humedad máxima para apagar el ventilador

#define BUZZERPIN 27     // Pin de señal del buzzer

#define GAS\_THRESHOLD 2000  // Umbral de detección de gas

#define PIRPIN 12        // Pin de señal del sensor PIR

#define LEDPIN1 4        // Pin de salida del LED 1

#define LEDPIN2 5        // Pin de salida del LED 2

#define PIN\_MICROPHONE 5 //Pin del SmallSound

// Configuración de RED y Broker

const char\* ssid = "OPPO Reno5 Lite";

const char\* password = "esmeralda25";

const char\* mqtt\_server = "192.168.71.135";

const char\* mqtt\_topic = "iot/temp";

#define BOT\_TOKEN "6039973965:AAG4YKk0\_y0bL3Swto-967W8Jo8Uh3nlCu4"

const unsigned long BOT\_MTBS = 1000; // mean time between scan messages

WiFiClientSecure secured\_client;

UniversalTelegramBot bot(BOT\_TOKEN, secured\_client);

unsigned long bot\_lasttime; // last time messages' scan has been done

//const int servoPin = SERVOPIN;

int servoStatus = 0;

WiFiClient wifi\_client;

PubSubClient mqtt\_client(wifi\_client);

DHT dht(DHTPIN, DHTTYPE);  // Inicialización del sensor DHT

Servo servo;               // Inicialización del servo

int buzzerState = LOW;     // Estado inicial del buzzer

int pirState = LOW;        // Estado inicial del sensor PIR

int ledState1 = LOW;       // Estado inicial del LED 1

int ledState2 = LOW;       // Estado inicial del LED 2

//telegram

void handleNewMessages(int numNewMessages)

{

  Serial.print("handleNewMessages ");

  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++)

  {

    String chat\_id = bot.messages[i].chat\_id;

    String text = bot.messages[i].text;

    String from\_name = bot.messages[i].from\_name;

    if (from\_name == "")

      from\_name = "Guest";

    if (text == "/ON")

    {

       servo.write(0);    // gira el servomotor a 0 grados

      delay(1000);         // espera 1 segundo

      servo.write(180);  // gira el servomotor a 180 grados

      delay(1000);

      servoStatus = 1;

      bot.sendMessage(chat\_id, "Servo is ON", "");

    }

    if (text == "/OFF")

    {

      servo.write(0);

      servoStatus = 0;

      bot.sendMessage(chat\_id, "Servo is OFF", "");

    }

    if (text == "/status")

    {

      if (servoStatus)

      {

        bot.sendMessage(chat\_id, "Servo is ON", "");

      }

      else

      {

        bot.sendMessage(chat\_id, "Servo is OFF", "");

      }

    }

    if (text == "/start")

    {

      String welcome = "Welcome to Universal Arduino Telegram Bot library, " + from\_name + ".\n";

      welcome += "This is Flash Led Bot example.\n\n";

      welcome += "/ledon : to switch the Led ON\n";

      welcome += "/ledoff : to switch the Led OFF\n";

      welcome += "/status : Returns current status of LED\n";

      bot.sendMessage(chat\_id, welcome, "Markdown");

    }

  }

}

void setup() {

  Serial.begin(9600);

  dht.begin();

  servo.attach(SERVOPIN);

  pinMode(BUZZERPIN, OUTPUT);

  pinMode(PIRPIN, INPUT);

  pinMode(LEDPIN1, OUTPUT);

  pinMode(LEDPIN2, OUTPUT);

  pinMode(PIN\_MICROPHONE,INPUT);

  servo.write(SERVOMIN);  // Configuración inicial del servo

  Serial.println("Conectando");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  secured\_client.setCACert(TELEGRAM\_CERTIFICATE\_ROOT); // Add root certificate for api.telegram.org

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

    Serial.print("Conectando...");

    Serial.println(".");

    delay(500);

  }

  Serial.println("Conexión WiFi establecida.");

  mqtt\_client.setServer(mqtt\_server, 1883);

  mqtt\_client.setCallback(callback);

}

void callback(char\* topic,byte\* payload, unsigned int length){

  payload[length]='\0';

  String val=String((char\*)payload);

  Serial.println(val);

}

void loop() {

  delay(2000);  // Retardo para la lectura del sensor

    if(!mqtt\_client.connected()){

  if(mqtt\_client.connect("ESP32")){

  Serial.println("Conexion exitosa");

  }

}

  float temp = dht.readTemperature();  // Lectura de la temperatura

  float hum = dht.readHumidity();      // Lectura de la humedad

  int gas = analogRead(2);            // Lectura del sensor MQ-2 de gas

  pirState = digitalRead(PIRPIN);      // Lectura del sensor PIR

  // Imprime los valores de temperatura, humedad, gas y PIR en la consola serial

  Serial.print("Temperatura: ");

  Serial.print(temp);

  Serial.print(" C   Humedad: ");

  Serial.print(hum);

  Serial.print("%   Gas: ");

  Serial.print(gas);

  Serial.print("   PIR: ");

  Serial.print(pirState);

  Serial.println();

  Serial.print("  Sound:");

  Serial.print(soundValue);

  // Si la temperatura está dentro del rango, enciende el ventilador

  if ( temp >= 20) {

    servo.write(180);

  } else {

    servo.write(90);  // Apaga el ventilador

  }

  //aqui

  //Condicional del SallSound

  if (soundValue == HIGH)

  {

    digitalWrite(LEDPIN2, HIGH);

    ledState2 = HIGH;

  }

  else

  {

    digitalWrite(LEDPIN2, LOW);

    ledState2 = LOW;

  }

  //1234567890º123456789

  // Si el sensor de gas detecta una concentración por encima del umbral, activa el buzzer

  if (gas >= GAS\_THRESHOLD) {

    digitalWrite(BUZZERPIN, HIGH);

    buzzerState = HIGH;

  } else {

    digitalWrite(BUZZERPIN, LOW);

    buzzerState = LOW;

  }

  if (pirState == HIGH) {

    digitalWrite(LEDPIN1, HIGH);

    ledState1 = HIGH;

  } else {

    digitalWrite(LEDPIN1, LOW);

    ledState1 = LOW;

  }

    char temp\_str[10];

  char hum\_str[10];

  char gas\_str[10];

  char pir\_str[10];

  char led\_str[10];

  sprintf(temp\_str, "%.2f", temp);

  sprintf(hum\_str, "%.2f", hum);

  sprintf(gas\_str, "%d", gas);

  sprintf(pir\_str, "%d", pirState);

  sprintf(led\_str, "%d", LEDPIN1); //

mqtt\_client.publish("temperatura", temp\_str);

mqtt\_client.publish("humedad", hum\_str);

mqtt\_client.publish("gas", pir\_str);

mqtt\_client.publish("servo", led\_str);

//mqtt\_client.publish("pir", pir\_str);

delay(1000);

   if (millis() - bot\_lasttime > BOT\_MTBS)

  {

    int numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

    while (numNewMessages)

    {

      Serial.println("got response");

      handleNewMessages(numNewMessages);

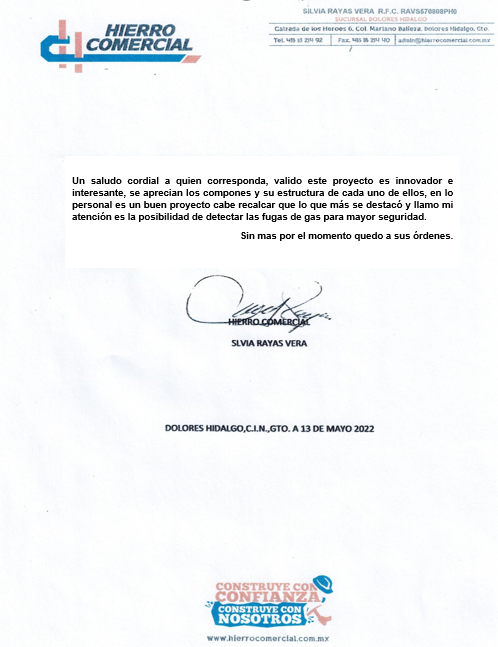
      numNewMessages = bot.getUpdates(bot.last\_message\_received + 1);

    }

    bot\_lasttime = millis();

  }

}

****