**ВИСНОВКИ**

В результаті виконання кваліфікаційної роботи була досягнута мета роботи, а саме була створена система пожежної сигналізації. Розроблена система цілком задовольняє вимоги щодо ціни та простоти реалізації.

Розроблена пожежна сигналізація має джерело резервного живлення, що забезпечує автономність та підвищує надійність спрацювання. У процесі створення сигналізації для мікроконтролера написані базові функції роботи з шлейфами сигналізації, функції відправки *SMS* повідомлень, перевірки датчиків та керування за допомогою веб-додатку.

Вся інформація з датчиків та подальші керуючі сигнали формує мікроконтролер *Teensy 3.2*, що є гарним вибором для такої розробки.

Застосування такої пожежної сигналізації підвищить безпеку та попередить можливе виникнення пожежі.

Програмний код перевірено на працездатність та підготовлено до використання.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ**

1. *GSM* сигналізація та охорона: http://www.vide.com.ua/alarm/gsm/

2. Охоронна система: https://ru.wikipedia.org/wiki/Технические\_средства\_охраны

3. Що таке *GSM* сигналізація: <http://www.vide.com.ua/documents/gsm-signalizacia-chto-eto/>

4. [Технічні засоби охоронно-пожежної сигналізації](https://ua-referat.com/Технічні_засоби_охоронно-пожежної_сигналізації): <https://ua-referat.com/Технічні_засоби_охоронно-пожежної_сигналізації>

5. *Teensy 3.2*, опис: <https://arduino.ua/prod977-teensy-3-2>

6. Підключення *Teensy*: https://www.pjrc.com/teensy/teensyduino.html

7. *SIM900,* опис: <http://www.kosmodrom.com.ua/prodlist.php?name=-sim900>

8*. ESP-8266*: https://esp8266.ru/esp8266-podkluchenie-obnovlenie-proshivki/

9. DC-DC перетворювач: https://uk.wikipedia.org/wiki/DC-DC\_перетворювач

10. *MQ-2* опис і характеристики: <https://arduino.ua/prod298-modyl-datchika-dima-mq-2>

11. *MQ-7* https://3d-diy.ru/wiki/arduino-datchiki/datchik-ugarnogo-gaza-mq7/

12. Датчик вогню інфрачервоний *KY-026*: <https://hcomp.ru/shop/ky026/>

13. *DHT11:* https://arduinomaster.ru/datchiki-arduino/datchiki-temperatury-i-vlazhnosti-dht11-dht22/

14. П’єзоелемент: <https://engineering-solutions.ru/ultrasound/-piezomaterials/>

15. Вибір модуль контролю заряду-розряду АКБ : https://3v3.com.ua/-product\_210.html

16. *GSM-модуль SIM800L*: повна інструкція з використання: codius.ru/articles/GSM\_модуль\_SIM800L\_часть\_1

17. *MediaTek MT6261* стаття та огляд: https://mediatek-club.ru/mediatek-mt6261

18. Охорона праці: <https://ohranatrud-ua.ru/razdely-kursovykh-i-diplomnykh-po-ot.html>

19. Штучне освітлення: https://ledeffect.com.ua/images/\_\_branding-/dbn2018.pdf

20. План евакуації: <https://www.sop.com.ua/article/215-qqq-17-m3-01-03-2017-plan-evakuats-pri-pojej>

21. Cистеми пожежної та охоронної сигналізації. Христич В.В., Дерев'-янко О.А., Бондаренко С.М., Антошкін О.А.: АПБУ МВС України

22. Охоронні сповіщувачі: relamaks.com/category/охоронна-сигналізація/

23. Протипожежна автоматика: https://youcontrol.com.ua/catalog/-company\_details/21747743/

**ДОДАТОК А**

**ПРИКЛАД ПРОГРАМИ ПОЖЕЖНОЇ GSM СИГНАЛІЗАЦІЇ**

---------------------------Код для прошивки МК Teensy------------------------

#include <SoftwareSerial.h>

#include "sms.h"

#include "call.h"

#include "DHT.h"

#include <SoftwareSerial.h>

SoftwareSerial ESPserial(0, 1);

SoftwareSerial mySerial(9, 10);

bool alarmState = false;

bool flaggas = false;

bool flagfire = false;

const int dhtPin = 2;

float tempValue;

const int mq2Pin = A0;

int mq2Value;

const int mq7Pin = A1;

int mq7Value;

const int firePin = A2;

int fireValue;

const int buzzer = 8;

const int greenLedPin = 3;

const int orangeLedPin = 4;

const int redLedPin = 5;

const int resetgsm = 13;

char nomer[] = "+380123456789";

const String alarmOffCommand = "0";

const String alarmOnCommand = "1";

const char smsOn[] = "Увімкнення системи!";

const char smsOff[] = "Вимкнення системи!";

const int smsBufferSize = 160;

const int gasThres = 600;

const int fireThres = 100;

const float tempThres = 28;

SMSGSM sms;

CallGSM call;

DHT dht(dhtPin, DHT22);

//Налаштування послідовного зв'язку між Arduino, GSM та Wi-Fi модулями

void setup() {

Wire.begin(8);

Wire.onReceive(receiveEvent);

Wire.onRequest(requestEvent);

Serial.begin(9600);

ESPserial.begin(9600);

pinMode(buzzer, OUTPUT);

pinMode(ledPin, OUTPUT);

pinMode(resetgsm, OUTPUT);

pinMode(firePin, INPUT);

pinMode(gasPin, INPUT);

mySerial.begin(115200);

mySerial.println("AT");

if (!mySerial.find("OK")) {

digitalWrite(resetgsm, HIGH);

delay(1000);

digitalWrite(resetgsm, LOW);

}

delay(2000);

while (1) {

mySerial.println("AT+COPS?");

if (mySerial.find("+COPS: 0")) break;

}

mySerial.println("AT+CMGF=1");

delay(100);

mySerial.println("AT+CSCS=\"GSM\"");

delay(100);

}

//Головна підпрограма

void loop() {

checkSms(); //Перевіряємо новоприбулі команди з GSM модулю

readEsp(); //Зчитуємо команди з веб-серверу

checkSensors(); //Зчитуємо значення датчиків та опрацьовуємо їх

sendToEsp(); //Відправляємо значення на веб сервер

delay(5000);

}

//Зчитуємо значення датчиків та опрацьовуємо їх

void checkSensors() {

mq2Value = analogRead(mq2Pin);

mq7Value = analogRead(mq7Pin);

fireValue = analogRead(firePin);

tempValue = dht.readTemperature();

if (mq2Value < gasThres && mq7Value < gasThres && fireValue < fireThres && tempValue >= tempThres && alarmState == true && state1 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчика температури. Висока вірогідність хибного спрацювання!");

state1 = true;

} else if (mq2Value < gasThres && mq7Value < gasThres && fireValue > fireThres && tempValue < tempThres && alarmState == true && state2 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчика вогню. Висока вірогідність хибного спрацювання!");

state2 = true;

} else if (mq2Value < gasThres && mq7Value < gasThres && fireValue > fireThres && tempValue > tempThres && alarmState == true && state3 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчиків вогню та температури. Вірогідність пожежі 50%");

state3 = true;

} else if (mq2Value < gasThres && mq7Value > gasThres && fireValue < fireThres && tempValue < tempThres && alarmState == true && state4 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчика вуглекислого газу. Висока вірогідність хибного спрацювання!");

state4 = true;

} else if (mq2Value < gasThres && mq7Value > gasThres && fireValue < fireThres && tempValue > tempThres && alarmState == true && state5 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчиків вуглекислого газу та температури. Вірогідність пожежі 50%");

state5 = true;

} else if (mq2Value < gasThres && mq7Value > gasThres && fireValue > fireThres && tempValue < tempThres && alarmState == true && state6 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчиків вуглекислого газу та вогню. Вірогідність пожежі 50%");

state6 = true;

} else if (mq2Value < gasThres && mq7Value > gasThres && fireValue > fireThres && tempValue > tempThres && alarmState == true && state7 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, HIGH);

tone(buzzer, 2780);

sms.sendSMS(nomer, "Спрацювання датчиків вуглекислого газу, вогню та температури. Висока вірогідність пожежи!");

state7 = true;

} else if (mq2Value > gasThres && mq7Value < gasThres && fireValue < fireThres && tempValue < tempThres && alarmState == true && state8 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчика диму. Висока вірогідність хибного спрацювання!");

state8 = true;

} else if (mq2Value > gasThres && mq7Value < gasThres && fireValue < fireThres && tempValue > tempThres && alarmState == true && state9 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчиків диму та температури. Вірогідність пожежі 50%");

state9 = true;

} else if (mq2Value > gasThres && mq7Value < gasThres && fireValue > fireThres && tempValue < tempThres && alarmState == true && state10 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчиків диму та вогню. Вірогідність пожежі 50%");

state10 = true;

} else if (mq2Value > gasThres && mq7Value < gasThres && fireValue > fireThres && tempValue > tempThres && alarmState == true && state11 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, HIGH);

tone(buzzer, 2780);

sms.sendSMS(nomer, "Спрацювання датчиків диму, вогню та температури. Висока вірогідність пожежи!");

state11 = true;

} else if (mq2Value > gasThres && mq7Value > gasThres && fireValue < fireThres && tempValue < tempThres && alarmState == true && state12 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, HIGH);

digitalWrite(redLedPin, LOW);

sms.sendSMS(nomer, "Спрацювання датчиків диму та вуглекислого газу. Вірогідність пожежі 50%");

state12 = true;

} else if (mq2Value > gasThres && mq7Value > gasThres && fireValue < fireThres && tempValue > tempThres && alarmState == true && state13 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, HIGH);

tone(buzzer, 2780);

sms.sendSMS(nomer, "Спрацювання датчиків диму, вуглекислого газу та температури. Висока вірогідність пожежи!");

state13 = true;

} else if (mq2Value > gasThres && mq7Value > gasThres && fireValue > fireThres && tempValue < tempThres && alarmState == true && state14 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, HIGH);

tone(buzzer, 2780);

sms.sendSMS(nomer, "Спрацювання датчиків диму, вуглекислого газу та вогню. Висока вірогідність пожежи!");

state14 = true;

} else if (mq2Value > gasThres && mq7Value > gasThres && fireValue > fireThres && tempValue > tempThres && alarmState == true state15 == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, HIGH);

tone(buzzer, 2780);

sms.sendSMS(nomer, "Увага! Виявлено пожежу!");

state15 = true;

}

if (mq2Value < gasThres && mq7Value < gasThres && fireValue < fireThres && tempValue < tempThres && alarmState == true) {

digitalWrite(greenLedPin, HIGH);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, LOW);

noTone(buzzer);

} else if (alarmState == false) {

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, LOW);

noTone(buzzer);

}

}

//Відправляємо значення на веб сервер

void sendToEsp() {

if (alarmState == true) {

request[0] = '1';

} else if (alarmState == false) {

request[0] = '0';

}

request[1] = String(mq2Value);

request[2] = String(mq7Value);

request[3] = String(fireValue);

request[4] = String(tempValue);

if (mq2Value < gasThres && mq7Value < gasThres && fireValue < fireThres && tempValue < tempThres) {

request[5] = '0';

} else if (mq2Value > gasThres || mq7Value > gasThres || fireValue > fireThres || tempValue > tempThres) {

request[5] = '1';

} else if (mq2Value > gasThres && mq7Value > gasThres && fireValue > fireThres && tempValue > tempThres) {

request[5] = '2';

}

sendToEsp = request[0] + request[1] + request[2] + request[3] + request[4] + request[5];

char tempArray[6];

sendToEsp.toCharArray(tempArray, 6);

ESPserial.write(tempArray);

}

//Перевіряємо новоприбулі команди з GSM модулю

void checkSms() {

char receivedSms = sms.IsSMSPresent(SMS\_UNREAD);

if (receivedSms) {

String smsContent = getSmsContent(receivedSms);

if (smsContent == alarmOffCommand) {

sms.sendSMS(nomer, smsOff);

alarmState = false;

resetFlags();

} else if (smsContent == alarmOnCommand) {

sms.sendSMS(nomer, smsOn);

alarmState = true;

resetFlags();

}

sms.DeleteSMS(receivedSms);

}

}

//Отримуємо вміст смс-повідомлень

String getSmsContent(char receivedSms) {

char smsBuffer[smsBufferSize];

sms.GetSMS(receivedSms, nomer, smsBuffer, sizeof(smsBuffer));

return String(smsBuffer);

}

//Cкидання значень датчиків

void resetFlags() {

noTone(buzzer);

digitalWrite(greenLedPin, LOW);

digitalWrite(orangeLedPin, LOW);

digitalWrite(redLedPin, LOW);

state1 = false;

state2 = false;

state3 = false;

state4 = false;

state5 = false;

state6 = false;

state7 = false;

state8 = false;

state9 = false;

state10 = false;

state11 = false;

state12 = false;

state13 = false;

state14 = false;

state15 = false;

}

//Зчитуємо команди з веб-серверу

void readEsp() {

char request;

String string;

if (ESPserial.available()) {

string = ESPserial.readString();

string.toCharArray(request, 1);

if (request == '0') {

alarmState = false;

} else if (request == '1') {

alarmState = true;

}

}

}

-----------------------------Код для прошивки ESP8266------------------------

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

#include <ESP8266mDNS.h>

const char \* ssid = "Wi-fi name";

const char \* password = "password";

char mq2Value[];

char mq7Value[];

char fireValue[];

char tempValue[];

char request;

String alarmState;

String alarmStatus;

WiFiServer server(80);

void setup() {

Serial.begin(9600);

delay(10);

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(1000);

Serial.print(".");

}

Serial.println("WiFi connected");

Serial.println("Server started");

Serial.print("Use this URL to connect: ");

Serial.print("http://");

Serial.print(WiFi.localIP());

Serial.println("/");

}

//Відображення значень на веб-сторінці

void loop() {

WiFiClient client = server.available();

if (!client) {

return;

}

Serial.println("new client");

String request = client.readStringUntil('\r');

Serial.println(request);

client.flush();

if (request.indexOf("/SYS=ON") != -1) {

Serial.print("ON");

request = "1";

Serial.write(request)

}

if (request.indexOf("/SYS=OFF") != -1) {

Serial.print("OFF");

request = "0";

Serial.write(request)

}

serialRequest();

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"/> <meta charset=\"utf-8\"><title>System info</title><style>button{color:red;padding: 10px 27px;}</style></head>");

client.println("<h1 style=\"text-align: center;font-family: Open sans;font-weight: 100;font-size: 20px;\">Інформаційна система протипожежної безпеки</h1><div>");

if (alarmState == "ON") {

client.println("<div style=\"text-align: center;width: 98px;color:white;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;\">Система ввімкнена</div>");

} else if (alarmState == "OFF") {

client.println("<div style=\"text-align: center;width: 98px;color:white;border: 5px solid #fff;padding: 10px 80px;background-color: #ec1212;margin: 0 auto;\">Система вимкнена</div>");

}

if (alarmStatus == "FINE") {

client.println("<div style=\"text-align: center;width: 98px;color:white;border: 5px solid #fff;padding: 10px 80px;background-color: #43a209;margin: 0 auto;\">Загроз немає</div>");

} else if (alarmStatus == "ERROR") {

client.println("<div style=\"text-align: center;width: 180px;color:white;border: 5px solid #fff;padding: 10px 38px;background-color: #ff8000;margin: 0 auto;\">Деякі датчики спрацювали, можливе хибне спрацювання</div>");

} else if (alarmStatus == "DANGER") {

client.println("<div style=\"text-align: center;width: 98px;color:white;border: 5px solid #fff;padding: 10px 80px;background-color: #ec1212;margin: 0 auto;\">Увага! Пожежа!</div>");

}

client.println("<p><center>Показники:</center>");

client.println("<br>");

client.println("<p><center>Датчик диму: ");

client.println(mq2Value);

client.println(" ppm</center>");

client.println("<p><center>Датчик вуглекислого газу: ");

client.println(mq7Value);

client.println(" ppm</center>");

client.println("<p><center>Датчик вогню: ");

client.println(fireValue);

client.println("</center>");

client.println("<p><center>Датчик температури: ");

client.println(tempValue);

client.println("&deg;C </center>");

client.println("<br>");

client.println("<center> Керування системою: </center>");

client.println("<div style=\"text-align: center;margin: 5px 0px;\"> <a href=\"/SYS=ON\"><button>Увімнути</button></a>&nbsp;<a href=\"/SYS=OFF\"><button>Вимнути</button></a></div>");

client.println("</html>");

delay(1);

Serial.println("Client disconnected");

}

//Отримання значень з плати

void serialRequest() {

char request[6];

String string;

if (Serial.available()) {

string = Serial.readString();

string.toCharArray(request, 8);

if (request[0] == '0') {

alarmState = "ON";

} else if (request[0] == '1') {

alarmState = "OFF";

}

mq2Value = request[1];

mq7Value = request[2];

fireValue = request[3];

temperature = request[4];

if (request[5] == '0') {

alarmState = "FINE";

} else if (request[5] == '1') {

alarmState = "ERROR";

} else if (request[5] == '2') {

alarmState = "DANGER";

}

}

}

**ДОДАТОК Б**

**ПРИНЦИПОВА СХЕМА ПІДКЛЮЧЕННЯ ЕЛЕМЕНТІВ СИГНАЛІЗАЦІЇ**

