**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 високу автономність та підвищує надійність спрацювання. У процесі створення *GSM*-сигналізації для мікроконтролера написані базові функції роботи з шлейфами сигналізації, функції відправки *SMS* повідомлень.

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

Застосування такої пожежної сигналізації підвищить безпеку власності 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. Що таке *GSM* сигналізація: http://www.vide.com.ua/documents/gsm-signalizacia-chto-eto/

3. Охоронна система: [https://ru.wikipedia.org/wiki/%D0%9E%D1%85%D-1%80%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0](https://ru.wikipedia.org/wiki/%D0%9E%D1%85%25D-1%80%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)

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

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

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

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

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

9. Опис мікроконтролерів *PIC16C7X*: http: // www. microchip.com.

10. Опис мікроконтролерів *Atmega*: http: // www. atmel.com.

11. Одно кристальний мікроконтролери *Microchip: PIC16C5X*. Під ред. А.Н. Владимирова. - М: ORMIX, 1996. - 96 с.

12. Пристрій мікроконтроллера ATmega328: http://robolive.ru/-mikrokontroller-atmega328-opisanie-xarakteristiki/

13. Плата Arduino Nano v 3.0 : розпіновка, схеми, драйвер: https://arduinomaster.ru/platy-arduino/plata-arduino-nano/#\_Arduino-Nano

14. Розпіновка процесорів atmega та atini: http://www.electronica52.in.ua/-proekty-arduino/raspinovki-processorov-atmega--i-atini

15. [https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D1%8B\_-%D0%](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D1%8B_-%D0%25)BA%D0%BE%D1%80%D0%BF%D1%83%D1%81%D0%BE%D0%B2

16. Бєлов А.В. - Мікроконтролери AVR в радіоаматорського практиці, 2007. – 352 с.

17. *Arduino Nano*: огляд і характеристики плати: https://arduinoplus.ru/arduino-nano/

18. *DC-DC* перетворювачі: http://electrik.info/main/praktika/1112-dcdc-preobrazovateli.html

19. *MediaTek MT6261* стаття та огляд:

[https://mediatek-club.ru/mediatek-mt6261](https://mediatek-club.ru/mediatek-mt6261 20)

[20](https://mediatek-club.ru/mediatek-mt6261 20). *GSM-модуль SIM800L*: повна інструкція з використання: <http://codius.ru/articles/GSM_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_SIM800L_%D1%87%D0%B0%D1%81%D1%82%D1%8C_1>

21. *GSM* та *GPRS* модулі: https://arduinomaster.ru/datchiki-arduino/gsm-gprs-modul-arduino/

**ДОДАТОК А**

**ПРИКЛАД ПРОГРАМИ ПОЖЕЖНОЇ GSM СИГНАЛІЗАЦІЇ**

#include <SoftwareSerial.h> //підключаємо бібліотеку для програмного Rx Tx

SoftwareSerial mySerial(2, 3); // RX, TX

//бібліотеки для роботи з смс і дзвінками

#include "sms.h"

#include "call.h"

//об'єкти класу для роботи з смс і дзвінками

SMSGSM sms;

CallGSM call;

bool alarmState = false;//стан сигналізації: true - ввімк, false - вимк

bool flaggas = false;

bool flagfire = false;

const int GasPin = A5; //пін для сенсора газу

int GasValue = 0; //змінна для зберігання стану датчика газу

const int FirePin = A4; //пін для датчика вогню

int FireValue = 0; //змінна для зберігання стану датчика вогню

const int buzzer = 8; //пін для п'єзовипромінювача

const int ledPin = 10;//пін для led

const int resetgsm = 9;//пін для resetgsm

char nomer[] = "+380123456789";//номер абонента

const String AlarmOffCommand = "0";//Команда зняття з охорони

const String AlarmOnCommand = "1";//Команда постановки на охорону

const char smsOgon[] = "UVAHA! SPRATSIUVANNIA DATCHYKA VOGNIU!";

const char smsGas[] = "UVAHA! SPRATSIUVANNIA DATCHYKA GAZU!";

const char smsOn[] = "POSTANOVKA NA OKHORONU!";

const char smsOff[] = "ZNIATO Z OKHORONY!";

const int SmsBufferSize = 160;// Розмір буфера SMS-повідомлення

const int sensorThres = 600;//Поріг зпрацювання датчика

void setup()

{

pinMode(buzzer, OUTPUT);

pinMode(ledPin, OUTPUT);

pinMode(resetgsm, OUTPUT);

pinMode(FirePin, INPUT);

pinMode(GasPin, INPUT);

mySerial.begin(115200);//швидкість порта

delay(2000);//час на ініціалізацію модуля

mySerial.println("AT");//чи є відповідь від модема

delay(100);

if (!mySerial.find("OK"))//якщо ні-ресет

{

digitalWrite(resetgsm, HIGH);

delay(1000);

digitalWrite(resetgsm, LOW);

}

delay(2000);//потрібно дочекатися включення модему і з'єднання з мережею

while(1)//чекаємо підключення модему до мережі

{

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

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

}

mySerial.println("AT+CMGF=1"); //режим кодування СМС - звичайний (для англ.)

delay(100);

mySerial.println("AT+CSCS=\"GSM\""); //режим кодування тексту

delay(100);

}

void loop()

{

CheckSms();

CheckSensor();

delay(3000);

}

void CheckSensor()

{

GasValue = analogRead(GasPin);//Зчитуємо значення з датчика газа

if ((GasValue >= sensorThres) && flaggas==false && alarmState==true)

{

digitalWrite(ledPin, HIGH);

tone(buzzer, 2780);//Сирена pin freq

sms.SendSMS(nomer, smsGas);//Відправляємо SMS-повідомлення про знаходження газу

sms.DeleteSMS(1);

flaggas=true;

}

FireValue = analogRead(FirePin);//Зчитуємо значення з датчика вогню

if ((FireValue >= sensorThres) && flagfire==false && alarmState==true)

{

digitalWrite(ledPin, HIGH);

tone(buzzer, 2780);//Сирена pin freq

sms.SendSMS(nomer, smsOgon);//Відправляємо SMS-повідомлення про знаходження вогню

sms.DeleteSMS(1);

flagfire=true;

}

}

void CheckSms()//Перевіряємо, чи прийшло SMS-повідомлення

{

char receivedSms = sms.IsSMSPresent(SMS\_UNREAD);//Перевіряємо непрочитанні SMS-повідомлення

if (receivedSms)//Якщо є непрочитані SMS-повідомлення, то...

{

String smsContent = GetSmsContent(receivedSms);//Отримуємо вміст непрочитаного SMS-повідомлення

if (smsContent == AlarmOffCommand)//Вимикаємо сигналізацію, якщо прийшла команда вимкнення

{

noTone(buzzer);//Вимикаємо п'єзолемент

digitalWrite(ledPin, LOW);

sms.SendSMS(nomer, smsOff);//Відправляємо SMS-повідомлення про вимкнення сигналізації

alarmState = false;

flaggas = false;

flagfire = false;

}

else if (smsContent == AlarmOnCommand)//Вмикаємо сигналізацію, якщо прийшла команда ввімкнення

{

noTone(buzzer);//Вимикаємо п'єзолемент

digitalWrite(ledPin, LOW);

sms.SendSMS(nomer, smsOn);//Відправляємо SMS-повідомлення про ввімкнення сигналізації

alarmState = true;

flaggas = false;

flagfire = false;

}

sms.DeleteSMS(receivedSms);//Видаляємо SMS-повідомлення з SIM-карти

}

}

String GetSmsContent(char receivedSms)//Отримуємо вміст SMS-повідомлення

{

char smsBuffer[SmsBufferSize];

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

return String(smsBuffer);

}

**ДОДАТОК Б**

**ПРИНЦИПОВА СХЕМА ПІДКЛЮЧЕННЯ ЕЛЕМЕНТІВ СИГНАЛІЗАЦІЇ**

