

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

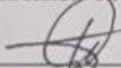
на тему:

**«СИСТЕМА ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ
ARDUINO»**

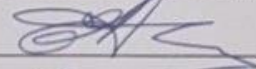
Виконав: студент 2 курсу, групи ІБС-23м
спеціальності 125 Кібербезпека та захист
інформації

 Максим КУТНЯК

Керівник: к. т. н., доцент каф. ЗІ

 Леонід КУПЕРШТЕЙН
«17» 12 2024 р.

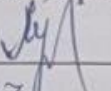
Опонент: к. т. н., доцент каф. ПЗ

 Олена КОВАЛЕНКО
«17» 12 2024 р.

Допущено до захисту

Завідувач кафедри ЗІ

д. т. н., проф.

 Володимир ЛУЖЕЦЬКИЙ
«17» 12 2024 р.

Вінниця ВНТУ – 2024 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Ступінь вищої освіти Магістр
Галузь знань – 12 Інформаційні технології
Спеціальність – 125 Кібербезпека та захист інформації
Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри ЗІ,

д. т. н., проф.

Володимир ЛУЖЕЦЬКИЙ

«18» 09 2024 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Кутняку Максиму Миколайовичу

1. Тема роботи: «Система охоронної сигналізації на основі платформи Arduino»
керівник роботи: Куперштейн Леонід Михайлович, к. т. н., доцент кафедри ЗІ,
затверджені наказом ректора ВНТУ від 17 вересня 2024 року №310.
2. Строк подання студентом роботи 17 грудня 2024 р.
3. Вихідні дані до роботи:
 - контроль несанкціонованого проникнення;
 - відображення стану системи на LCD-дисплеї;
 - віддалений доступ до системи;
 - предметна область – контроль та інформування про несанкціоноване проникнення;
 - мова програмування – об'єктно-орієнтована;
 - кількість параметрів мікроклімату не менше 2;
 - режим енергозаощадження;
 - платформа Arduino;
 - вбудовані датчики температури, вологості та якості повітря.
4. Зміст текстової частини: Вступ. Аналіз предметної області. Проектування програмно-апаратної системи охоронної сигналізації на основі платформи Arduino. Висновок. Список використаної літератури.
5. Перелік графічного матеріалу.
Загальна структурна схема. Монтажна схема. UML діаграма комунікації. UML діаграма станів. Схема алгоритму роботи контролера з WiFi. Схема алгоритму роботи головного контролера

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Куперштейн Л. М., доц. кафедри ЗІ	 10.09.24	 15.09.24
2	Куперштейн Л. М., доц. кафедри ЗІ	 16.09.24	 25.09.24
3	Куперштейн Л. М., доц. кафедри ЗІ	 14.10.24	 10.11.24
4	Ратушняк О. Г., доц. кафедри ЕПВМ	 11.11.24	 17.11.24

7. Дата видачі завдання – 1 вересня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2024 - 10.09.2024	
2	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	10.09.2024 - 15.09.2024	
3	Науково-технічне обґрунтування	16.09.2024 - 22.09.2024	
4	Аналіз предметної області та формування вимог до програмного засобу	23.09.2024 - 29.09.2024	
5	Розробка архітектури програмного засобу	30.09.2024 - 12.10.2024	
6	Програмна реалізація засобу	14.10.2024 - 02.11.2024	
7	Розробка методики тестування розробленого засобу	3.11.2024 - 10.11.2024	
8	Застосування методики тестування розробленого засобу	11.11.2024 - 17.11.2024	
9	Розробка розділу економічного обґрунтування доцільності розробки	18.11.2024 - 21.11.2024	
10	Аналіз виконання ТЗ, висновки	22.11.2024 - 24.11.2024	
11	Оформлення пояснювальної записки	25.11.2024 - 29.11.2024	
12	Перевірка магістерської роботи на наявність текстових запозичень	30.10.2024 - 02.12.2024	
13	Попередній захист та доопрацювання МКР	03.12.2024 - 14.12.2024	
14	Представлення МКР до захисту, рецензування	15.12.2024 - 17.12.2024	
15	Захист МКР	17.12.2024 - 20.12.2024	

Студент  Максим КУТ
Керівник роботи  Леонід КУПЕРШТ

АНОТАЦІЯ

УДК 004.056

Кутняк М. Система охоронної сигналізації на основі платформи Arduino. Магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека та захист інформації, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2024. 98 с.

Укр. мовою. Бібліогр.: 47 назв; рис.: 37; табл.: 1.

Магістерська кваліфікаційна робота присвячена вдосконаленню системи охоронної сигналізації на основі платформи Arduino. Підготовлено науково-дослідне та техніко-економічне обґрунтування доцільності досліджень. У роботі здійснено аналіз існуючих систем охоронних сигналізації і обґрунтовано вибір платформи для реалізації. Розроблено структурну схему, монтажну схему охоронної сигналізації. Розроблено UML-діаграми функціонування системи охоронної сигналізації. Програмно реалізовано та здійснено апаратне проектування системи.

Ілюстративна частина складається з 7 плакатів з демонстрацією результатів моделювання і проведених досліджень.

В економічному розділі оцінено витрати на розробку.

Ключові слова: система охоронної сигналізації, сигналізація, Arduino, мікроконтролери, безпека, геркон, моніторинг параметрів мікроклімату, датчик руху, датчик Холла, Wi-Fi модуль, SMS-сповіщення, GSM-модуль, SIM-карта, I2C дисплей.

ABSTRACT

UDC 004.56

Kutnyak M. Security alarm system based on the Arduino platform. Master's qualification work in the specialty 125 - Cybersecurity and information protection, educational program - Security of information and communication systems. Vinnytsia: VNTU, 2024. 99 p.

In Ukrainian. Bibliography: 47 titles; Fig.: 37; Table: 1.

The master's qualification work is devoted to the improvement of the security alarm system based on the Arduino platform. A scientific research and feasibility study of the research has been prepared. The work analyzes existing security alarm systems and justifies the choice of the platform for implementation. A structural diagram and a wiring diagram of the security alarm have been developed. UML diagrams of the functioning of the security alarm system have been developed. The system has been implemented and hardware design has been carried out.

The illustrative part consists of 7 posters demonstrating the results of modeling and research.

The economic section estimates the development costs.

Keywords: security alarm system, alarm, Arduino, microcontrollers, security, reed switch, monitoring of microclimate parameters, motion sensor, Hall sensor, Wi-Fi module, SMS notification, GSM module, SIM card, I2C display.

ЗМІСТ

ВСТУП.....	7
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ ТЕМИ ДОСЛІДЖЕННЯ.....	10
1.2 ОГЛЯД ВІДОМИХ РЕАЛІЗАЦІЙ ОХОРОННИХ СИГНАЛІЗАЦІЙ.....	12
1.3 ФОРМАЛІЗАЦІЯ ВИМОГ ТА ПОСТАНОВКА ЗАДАЧІ.....	20
1.4 ВИСНОВКИ	26
2 ПРОЕКТУВАННЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO	28
2.1 СТРУКТУРНА СХЕМА СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO.....	28
2.2 РОЗРОБКА МОНТАЖНОЇ СХЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO.....	31
2.3 РОЗРОБКА UML-ДІАГРАМ ФУНКЦІОНУВАННЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ	35
2.4 РОЗРОБКА АЛГОРИТМІВ ФУНКЦІОНУВАННЯ ПРОГРАМНО-АПАРATНОЇ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO	38
2.5 АНАЛІЗ ПОТЕНЦІЙНИХ ЗАГРОЗ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ	42
2.6 ТЕОРЕТИКО-МНОЖИННА МОДЕЛЬ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ	45
2.7 ВИСНОВКИ	48
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ПЛАТФОРМИ ARDUINO.....	49
3.1 ОБГРУНТУВАННЯ ВИБОРУ ПЛАТФОРМ ДЛЯ РОЗРОБКИ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ	49
3.2 ОСОБЛИВОСТІ СЕРЕДОВИЩА РОЗРОБКИ ARDUINO IDE	53
3.3 ПРОГРАМНА ТА АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ПЛАТФОРМИ ARDUINO	55

3.4	ТЕСТУВАННЯ РОБОТИ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	65
3.5	ВИСНОВОК	72
4	ЕКОНОМІЧНА ЧАСТИНА	73
4.1	ПРОВЕДЕННЯ КОМЕРЦІЙНОГО АУДИТУ НАУКОВО-ТЕХНІЧНОЇ РОЗРОБКИ.....	73
4.2	РОЗРАХУНОК ВИТРАТ НА ЗДІЙСНЕННЯ НАУКОВО-ДОСЛІДНОЇ РОБОТИ.....	77
4.2.1	Витрати на оплату праці	77
4.2.2	Відрахування на соціальні заходи	79
4.2.3	Амортизація програмних засобів	80
4.2.4	Сировина та матеріали	81
4.2.5	Паливо та енергія для науково-виробничих цілей	83
4.2.6	Інші витрати.....	83
4.2.7	Загальні витрати	84
4.3	РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ НАУКОВО-ТЕХНІЧНОЇ РОЗРОБКИ ПРИ ЇЇ МОЖЛИВІЙ КОМЕРЦІАЛІЗАЦІЇ ПОТЕНЦІЙНИМ ІНВЕСТОРОМ.....	84
4.4	ВИСНОВКИ	88
	ВИСНОВКИ	90
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	91
	ДОДАТОК А ПРОТОКОЛ ПЕРЕВІРКИ . ERROR! BOOKMARK NOT DEFINED.	
	ДОДАТОК Б КОД ПРОГРАМИ ARDUINO NANO.....	8
	ДОДАТОК В КОД ПРОГРАМИ ESP8266.....	15
	ДОДАТОК Г АКТ ВПРОВАДЖЕННЯ.....	19
	ДОДАТОК Д ІЛЮСТРАТИВНА ЧАСТИНА	24

ВСТУП

Сьогодні цифровізація життя сприяє перенесенню все більшої кількості даних в електронну форму, що впливає на різні сфери діяльності, включаючи системи безпеки та охорони. З розвитком технологій охоронні системи стають все більш інтелектуальними та здатні обробляти велику кількість інформації в режимі реального часу. Програмне забезпечення, яке використовується в таких системах, виконує роль основної ланки для обробки даних різного рівня важливості, що підвищує вимоги до забезпечення їх захищеності.

При створенні системи охоронної сигналізації на основі платформи Arduino важливо гарантувати фізичну безпеку. До таких даних відносяться сигнали від датчиків, керуючі команди та внутрішня інформація системи, яка формується, зберігається та використовується під час роботи.

Актуальність. Забезпечення безпеки людей та майна завжди було й залишається одним з найважливіших завдань суспільства. Охоронні сигналізації (ОС) відіграють ключову роль у попередженні незаконного проникнення та інших надзвичайних ситуацій. Традиційні ОС, як правило, ґрунтуються на дорогих та складних компонентах, що робить їх доступними не для всіх. Цей недолік робить актуальним пошук альтернативних рішень, які б поєднували в собі ефективність, доступність та простоту використання.

Мікроконтролери Arduino пропонують себе як перспективну платформу для розробки ОПС нового покоління. Завдяки своїй доступності, простоті програмування та гнучкості, Arduino робить можливим створення надійних та економічних систем сигналізації, які можуть бути адаптовані до різноманітних потреб.

Однією з таких критичних областей застосування кібербезпеки є проектування систем безпеки, зокрема охоронно сигналізацій. Зростаюча складність сучасних будівель, споруд та інфраструктурних об'єктів вимагає вдосконалення систем безпеки, що забезпечує реагування на небезпеку в реальному часі та надійний захист користувачів. Водночас, зростаюча кількість пожежних та кібератак на інфраструктуру додає складності до цієї проблеми.

Проектування охоронно систем на базі мікроконтролерів Arduino представляє собою перспективний шлях для забезпечення ефективного контролю та реагування на потенційні загрози. Arduino, як відомий відкритий апаратний та програмний засіб, дозволяє розробникам створювати кастомізовані рішення з високим рівнем гнучкості та доступності. Його використання в охоронних системах дозволяє створювати інтегровані, ефективні та стійкі до кібератак рішення.

Об'єкт дослідження є процеси забезпечення захисту приміщення від загроз як навмисного, так і ненавмисного характеру.

Предметом дослідження методи та засоби для приміщення від загроз як навмисного, так і ненавмисного характеру.

Метою дослідження є покращення захищеності приміщення від загроз несанкціонованого проникнення та техногенних загроз за рахунок розробки та впровадження охороної сигналізації.

Задачі дослідження:

1. Обґрунтувати доцільність розробки системи охоронної сигналізації на основі платформи Arduino;
2. Розглянути відомі реалізації охоронних сигналізацій.
3. Спроекувати систему охоронної сигналізації на основі платформи Arduino;
4. Програмно реалізувати систему охоронної сигналізації на основі платформи Arduino;
4. Виконати тестування програми та провести аналіз отриманих результатів.
5. Розробити інструкцію користувача.

Наукова новизна полягає у розробці моделі системи охоронної сигналізації на основі платформи Arduino для забезпечення захищеності приміщення на основі багаторівневого моніторингу, управління, сповіщення.

Практична цінність дослідження полягає в розробці програмно-апаратного засобу на основі платформи Arduino, який дозволяє покращити захищеність як житлових, так промислових приміщень за рахунок розширених можливостей керування та сповіщення.

Окремі результати магістерської кваліфікаційної роботи було впроваджено у діяльність ТОВ «Тірас-12», що підтверджується виданим Актом впровадження [Додаток Б].

Окремі результати роботи доповідались на V Міжнародній науково-практичній конференції «Інформаційна безпека та інформаційні технології» (Львівський державний університет безпеки життєдіяльності, 2024 р.) опубліковано тези у збірнику матеріалів конференції [1].

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування доцільності теми дослідження

Відомо, що охорона приміщень має велике значення для комфорту, здоров'я людей. Забезпечення інформування про несанкціоноване проникнення та пожежу в приміщенні є важливою задачею.

У сучасному світі, де безпека є пріоритетом для приватних осіб, підприємств та державних установ, системи охоронної сигналізації набувають все більшої актуальності. Зростання кількості загроз несанкціонованого доступу, крадіжок, вандалізму та інших злочинів вимагає застосування передових технологій для захисту об'єктів будь-якого рівня складності. Системи охоронної сигналізації дозволяють мінімізувати ризики, пов'язані з проникненням на територію, а також оперативно інформувати про загрози, що значно підвищує рівень безпеки.

Обґрунтування актуальності даної теми зумовлене, зокрема, швидким розвитком технологій у сфері безпеки, що включає вдосконалення технологій, інтеграцію з мережами Інтернету речей (IoT), а також можливості віддаленого моніторингу. Сучасні системи охоронної сигналізації дозволяють здійснювати постійний контроль об'єкта, управління доступом.

Крім того, системи охоронної сигналізації стають важливим інструментом для управління об'єктом, оскільки вони надають можливості комплексного підходу до забезпечення безпеки. Поєднання охоронної сигналізації з системами контролю доступу та іншими засобами безпеки дозволяє значно підвищити ефективність захисту, забезпечуючи координацію дій персоналу охорони у надзвичайних ситуаціях. Наприклад, такі системи можуть вчасно оповістити про вторгнення, задимлення, пожежу або інші небезпечні фактори, що дозволяє уникнути шкоди майну та здоров'ю людей. На рис 1.1 показана динаміка справ про хатні крадіжки 2021-2023 роках.

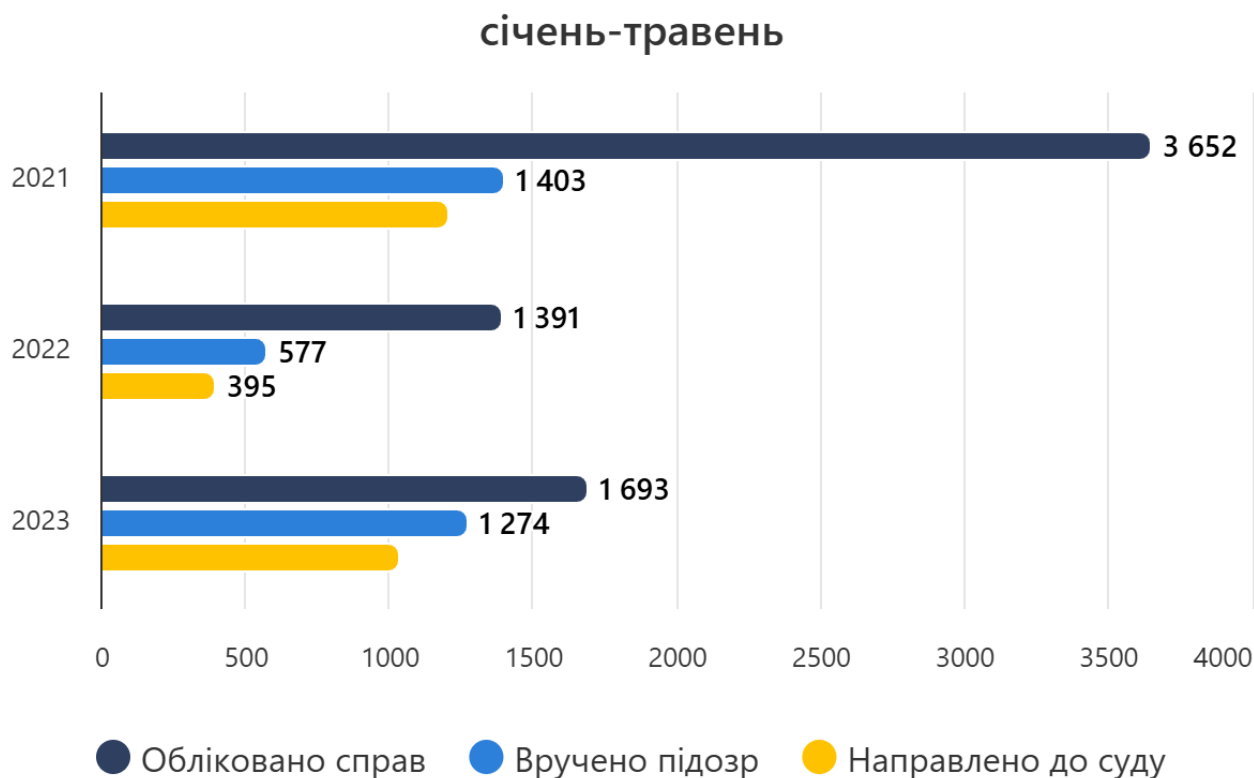


Рисунок 1.1 – Динаміка справ про хатні крадіжки 2021-2023 роках

За п'ять місяців 2023 року було відкрито 1693 провадження за фактами квартирних крадіжок (ст. 185 Кримінального Кодексу України). Це — на 22% більше, ніж за аналогічний проміжок часу в минулому році, але водночас більш ніж удвічі менше, ніж протягом січня-травня 2021 року. В середньому цього року на місяць реєструвалося 340 таких кримінальних проваджень, у 2022 році — 290, а у 2021 році — 670. При цьому, правоохоронці стали частіше «ловити» ймовірних квартирних крадіїв. У 2023 році частка справ, за якими потенційним зловмисникам було вручено підозри, складає 75%. Для порівняння, у 2021 році підозри отримали фігуранти 38% справ. У свою чергу, в січні-травні 2022 року цей показник становив 41%. Суттєво збільшується й кількість справ, які після розслідування були передані на розгляд суддів. Так, станом на кінець травня поточного року їх вже нараховувалося 1033, що складає 61% від усіх облікованих проваджень. Разом з тим, у 2021 році частка таких справ складала 33%, а у 2022 році — 28% [2].

У контексті підвищення вимог до безпеки в суспільстві, а також прагнення до зниження ризиків, пов'язаних з кримінальними загрозами та надзвичайними

ситуаціями, тема дослідження систем охоронної сигналізації є надзвичайно актуальною та має високу практичну цінність. Вона дозволяє не лише розширити наукові знання у сфері безпеки, але й створювати нові, більш досконалі технічні рішення, здатні забезпечити максимальний рівень захисту для всіх категорій об'єктів.

Система охоронної сигналізації є комплексом технічних засобів та програмного забезпечення, призначених для забезпечення захисту об'єктів від несанкціонованого доступу, проникнення та інших загроз [3]. Основними компонентами таких систем є датчики, виконавчі пристрої та блоки управління, які взаємодіють для забезпечення своєчасного виявлення та інформування про тривоги.

1.2 Огляд відомих реалізацій охоронних сигналізацій

Системи охоронної сигналізації пройшли значний шлях розвитку, і сьогодні існує безліч реалізацій, що забезпечують ефективний захист різних типів об'єктів. Вони варіюються за технологічними підходами, функціоналом, можливостями інтеграції, а також за складністю установки та обслуговування. Сучасні рішення включають як прості системи для приватного використання, так і високотехнологічні комплекси для захисту великих промислових та урядових об'єктів. У цьому розділі розглянемо основні типи охоронних сигналізацій і їх відомі реалізації на ринку [5]. На сьогодні на ринку охоронних сигналізацій є багато пропозицій від різних вітчизняних і закордонних виробників.

Сьогодні найкращими виробниками охоронних сигналізацій за підсумками аналізу пропозицій ринку належать такі системи.

TIRAS Technologies: Український виробник систем безпеки та протипожежних систем. TIRAS Technologies - це українська компанія, яка з 1991 року розробляє та виробляє професійні системи безпеки. Їх обладнання використовується для захисту сотень тисяч об'єктів в Україні та за її межами.

Серед приладів фірми Tiras є окремо лінійка прилади приймально-контрольні охоронні (ППК) Orion NOVA XS / S / M / L [5].

На рис. 1.2 зображений зовнішній вигляд охоронної сигналізації компактної моделі серії охоронних систем на 4 дротові зони/датчика.



Рисунок 1.2 – Зовнішній вигляд ППК Orion NOVA S

На рис. 1.3 зображений зовнішній вигляд прилад приймально-контрольний пожежний на 4 неадресні зони - Тірас-4П.



Рисунок 1.3 – Зовнішній вигляд ППКП Тірас-4П

Jablotron - чеський виробник, який пропонує широкий спектр ОПС для будь-яких потреб, від простих систем для дому до складних систем для комерційних

об'єктів. Jablotron відомий своїми інноваційними технологіями та високою якістю продукції [6]. На рис. 1.4 зображений зовнішній вигляд охоронної системи Jablotron.



Рисунок 1.4 – Зовнішній вигляд охоронної системи Jablotron

Paradox - канадський виробник, який пропонує широкий спектр надійних та економічних охоронних систем. Paradox - це популярний вибір для власників будинків та малого бізнесу [7]. На рис. 1.5 зображений зовнішній вигляд охоронної системи Paradox.



Рисунок 1.5 – Зовнішній вигляд охоронної системи Paradox

DSC: Американський виробник, який пропонує широкий спектр ОПС, включаючи як дротові, так і бездротові системи. DSC відомий своїми простими у використанні системами та надійною підтримкою клієнтів [8]. На рис. 1.6 зображений зовнішній вигляд охоронної системи DSC.



Рисунок 1.6 – Зовнішній вигляд охоронної системи DSC

Honeywell - американський виробник, який пропонує широкий спектр ОПС, включаючи як прості системи для дому, так і складні системи для комерційних об'єктів. Honeywell відомий своїми інноваційними технологіями та високою якістю продукції [9]. На рис. 1.7 зображений зовнішній вигляд охоронної системи Honeywell.



Рисунок 1.7 – Зовнішній вигляд охоронної системи Honeywell

Bosch - Німецький виробник, який пропонує широкий спектр ОПС, включаючи як прості системи для дому, так і складні системи для комерційних об'єктів. Bosch відомий своїми інноваційними технологіями та високою якістю продукції [10]. На рис. 1.8 зображений зовнішній вигляд охоронної системи Bosch.



Рисунок 1.8 – Зовнішній вигляд охоронної системи Bosch

Satel – польський виробник, що спеціалізується на системах охоронної сигналізації, системах контролю доступу та розумного дому. Satel відомий своїми надійними та гнучкими системами, які підходять як для малих об'єктів, так і для великих комплексів. Серія AVAX забезпечує бездротове рішення з високим рівнем захисту та широкими можливостями інтеграції з іншими системами безпеки та автоматизації [11]. На рис. 1.9 зображений зовнішній вигляд охоронної системи Satel.



Рисунок 1.9 – Зовнішній вигляд охоронної системи Satel

U-Prox – український виробник, що спеціалізується на розробці систем контролю доступу та сигналізації. Компанія пропонує сучасні рішення з використанням RFID-технологій, а також системи інтеграції з мобільними додатками та бездротовими мережами. Системи U-Prox підходять як для приватних будинків, так і для великих підприємств, забезпечуючи зручне управління та моніторинг у реальному часі [12]. На рис. 1.10 зображений зовнішній вигляд охоронної системи U-Prox.



Рисунок 1.10 – Зовнішній вигляд охоронної системи U-Prox

Hikvision – китайський виробник, лідер у сфері відеоспостереження та рішень для охоронних систем. Крім камер і відеореєстраторів, Hikvision пропонує широкий асортимент охоронних сигналізацій і систем контролю доступу, які легко інтегруються з відеоспостереженням. Продукція відома своїм функціоналом, високою надійністю та доступністю для різних масштабів об'єктів – від будинків до великих установ. Hikvision включає в себе комплексні рішення, які призначені для забезпечення ефективної безпеки об'єктів різного напрямку: відеореєстратори, камери відеоспостереження, контроль доступу, домофони і різне додаткове устаткування [13]. На рис. 1.11 зображений зовнішній вигляд охоронної системи Hikvision.

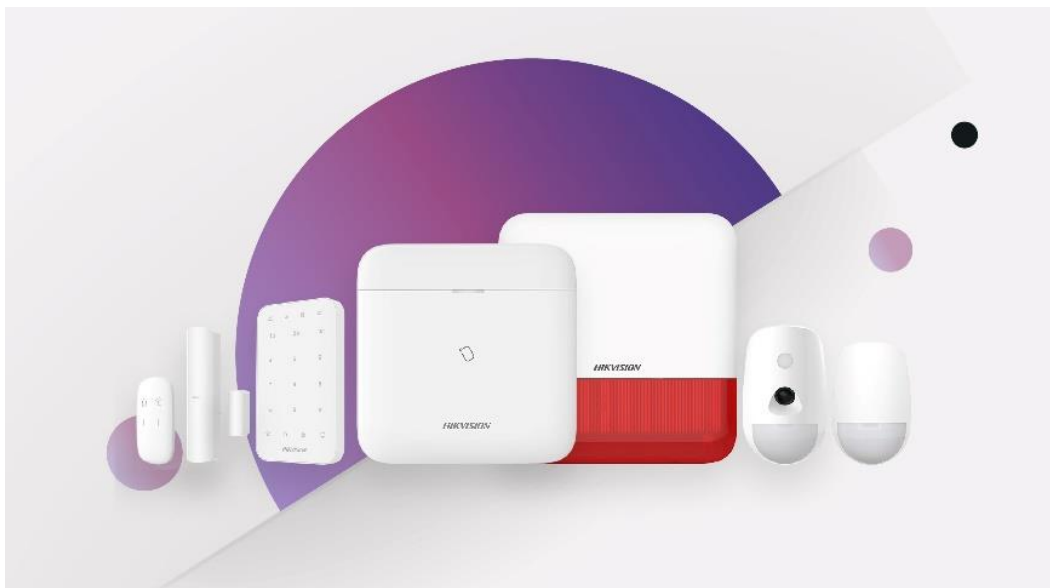


Рисунок 1.11 – Зовнішній вигляд охоронної системи Hikvision

Ајах – український виробник бездротових охоронних систем, які відрізняються сучасним дизайном, високою функціональністю та простотою встановлення. Ајах пропонує широкий спектр продуктів для захисту від проникнення, пожежної безпеки та контролю домашніх пристроїв. Компанія відома інноваційними рішеннями, зокрема використанням протоколу зв'язку Jeweller, що забезпечує захист від перешкод і надійність передачі сигналу [14]. На рис. 1.12 зображений зовнішній вигляд охоронної системи Ајах.



Рисунок 1.12 – Зовнішній вигляд охоронної системи Ајах

Для наочного подання, результати порівняння деяких систем охоронної сигналізації наведені в табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика систем охоронної сигналізації

Характеристика \ Модель	Tiras (Опiон NOVA S)	Satel	U- Prox	Hikvision	Ajax	Jablotron (JA-100)	Paradox (SP- 6000)	DSC (PowerSeries NEO)	Honeywell (Vista 20P)	Bosch (Solution 2000)
Контроль відчинення дверей	+	+	+	+	+	+	+	+	+	+
Контроль руху	+	+	+	+	+	+	-	-	+	+
Інформування SMS	+	+	+	+	+	+	+	+	+	+
Контроль затоплення	+	+	+	+	+	+	-	+	+	+
Вимірювання відносної вологості	-	-	-	+	+	-	-	-	+	+
Вимірювання рівня CO ₂	-	-	-	-	-	-	-	-	-	-
Вимірювання температури	+	+	+	+	+	+	+	+	+	+
Індикація показників вимірювання	+	+	+	+	+	+	+	+	+	+
Діапазон зон захисту	До 32 зон	До 48 зон	До 64 зон	До 32 зон	До 150 зон	До 120 зон	До 32 зон	До 128 зон	До 48 зон	До 8 зон
Інтеграція з мобільним додатком	+	+	+	+	+	+	-	+	+	+
Підтримка віддаленого моніторингу	+	+	+	+	+	+	-	+	+	+
Живлення резервне (акумулятор)	+	+	+	+	+	+	+	+	+	+
Захист від саботажу	+	+	+	+	+	+	+	+	+	+
Підтримка розумного дому (IoT)	-	+	-	+	+	+	-	-	+	-
Захист від глушіння сигналу	+	+	+	+	+	+	-	+	+	+

З табл. 1.1 видно, що існуючі прилади охоронних сигналізацій не в повному обсязі забезпечують підключення датчиків руху, вимірювання якості та відносної вологості повітря в приміщенні. В розроблюваній програмно-апаратній системі врахованим буде контроль відчинення дверей, контроль руху, інформування про постановку, зняття з-під охорони, тривоги, інформування в SMS, також вимірювання температури, відносної вологості, CO₂ та індикація показників вимірювання.

1.3 Формалізація вимог та постановка задачі

Сучасні системи охоронної сигналізації, такі як Tiras, Jablotron, Paradox, DSC, Honeywell, та Bosch, пропонують широкий функціонал, включаючи контроль відчинення дверей, інформування через SMS, моніторинг руху, захист від саботажу, а також можливість дистанційного моніторингу. Вони, безперечно, задовольняють базові потреби з безпеки, але не завжди підходять для адаптивного налаштування під конкретні вимоги, зокрема для невеликих чи спеціалізованих об'єктів, які потребують особливих налаштувань. Наприклад, у невеликих приміщеннях, таких як приватні будинки або невеликі офіси, часто необхідна система, що поєднує низьку вартість із можливістю індивідуального налаштування. У таких випадках готові комерційні рішення можуть виявитися надмірно складними або дорогими. Крім того, у спеціалізованих об'єктах, таких як лабораторії чи склади з особливими вимогами до безпеки, може знадобитися інтеграція додаткових сенсорів підвищеної точності (наприклад, для моніторингу температури, вологості чи концентрації газів), яку не завжди легко реалізувати з використанням стандартних систем.

Під час аналізу ринка виявлено такі обмеження та недоліки існуючих систем:

- Гнучкість і адаптивність: Більшість комерційних охоронних систем, хоча й пропонують розширений функціонал, мають обмежені можливості щодо індивідуального налаштування та розширення. Наприклад, системи Honeywell та Bosch розраховані на стандартизоване використання без глибокої інтеграції зі

сторонніми датчиками чи функціями, специфічними для об'єкта. Це може бути проблемою для об'єктів із нестандартною архітектурою або унікальними умовами експлуатації.

- Інтеграція з іншими технологіями: Технології на основі IoT і штучного інтелекту (як, наприклад, у Jablotron та DSC) лише частково реалізовані в більшості доступних на ринку систем. Для багатьох систем, таких як Paradox і Tiras, інтеграція з іншими "розумними" системами будинку є обмеженою або взагалі відсутньою, що знижує їхню ефективність у комплексній системі управління будинком. Для специфічних завдань, як-от моніторинг газів, рівня вологості або температури у промислових приміщеннях, комерційні рішення також не завжди мають достатній функціонал.

- Вартість: Брендіві системи, зокрема від Honeywell та DSC, мають високу вартість, що включає як початкову інвестицію, так і витрати на обслуговування. Для багатьох малих та середніх об'єктів, де не потрібна надмірна складність або де бюджет обмежений, таке рішення стає економічно недоцільним.

- Залежність від закритих протоколів: Багато виробників використовують власні протоколи зв'язку та обмежують можливості для інтеграції з іншими пристроями або платформами, що значно ускладнює налаштування і модернізацію системи. Наприклад, у системах Bosch та Honeywell складно додавати додаткові датчики або пристрої, що не є частиною екосистеми виробника.

Також, виявлені недоліки в існуючих системах:

- Гнучке налаштування зон контролю для об'єктів із нестандартним плануванням (наприклад, склади з безліччю входів, вентиляційних отворів, зон зберігання).

- Контроль навколишніх умов (температура, вологість, наявність диму чи газів) — стандартні охоронні системи рідко підтримують ці датчики або підтримують їх у обмеженому обсязі.

- Дистанційне налаштування та модернізація — важлива функція для віддалених об'єктів, зокрема без можливості доступу технічного персоналу, наприклад, на дачах чи в місцях, що підлягають сезонному захисту.

- Інтеграція з іншими технологіями безпеки та IoT — для оптимізації роботи охоронної системи на великих підприємствах потрібна інтеграція з системами контролю доступу, камерами, освітленням, датчиками та іншими підсистемами.

Доцільність розробки власної системи на основі платформи Arduino

З огляду на обмеження та недоліки комерційних систем, розробка охоронної системи на основі платформи Arduino може стати ефективним і економічно обґрунтованим рішенням. Використання платформи Arduino дозволить створити більш гнучку та адаптивну систему, що матиме наступні переваги:

- Масштабованість та налаштування — Arduino дозволяє легко додавати чи видаляти датчики та інші компоненти, відповідно до специфіки об'єкта. Можна налаштувати контроль зон захисту на найдетальнішому рівні, що підходить для будь-яких нестандартних умов.

- Економічність — Arduino є більш доступним варіантом, ніж брендові системи охорони. Компоненти, зокрема датчики руху, температури, диму, є порівняно дешевими, що дозволяє створити повнофункціональну систему за меншу вартість.

- Інтеграція з іншими системами — Arduino підтримує протоколи зв'язку (як-от Wi-Fi, Bluetooth, LoRa), що полегшує інтеграцію з іншими IoT-пристроями та дозволяє розробити систему "розумного дому". Можна підключити камери, сирени, системи оповіщення, навіть освітлення, і все це буде працювати як єдиний механізм.

- Можливість віддаленого моніторингу та керування. За допомогою додаткових модулів, таких як ESP8266 або ESP32, можна додати функції Wi-Fi, що забезпечать віддалений доступ до системи охорони через мобільний додаток або веб-інтерфейс. Це дозволяє користувачам контролювати стан об'єкта та отримувати сповіщення в режимі реального часу.

- Адаптивність до змін у середовищі — платформа Arduino дозволяє підключати широкий спектр датчиків, що дозволить створити більш комплексну систему з урахуванням конкретних умов об'єкта, таких як контроль температури, вологості, виявлення диму чи газу. Це особливо важливо для об'єктів, де існує ризик не лише проникнення, але й інших небезпек (наприклад, пожежі, витоків

газу). На основі зібраних даних, програмно-апаратна система має інформувати про несанкціоноване проникнення та показники мікроклімату в приміщенні.

Розроблена система обов'язково повинна виконувати такі функції:

- а) система має контролювати відкриття дверей та інформувати коритувача звуковою індикацією;
- б) система має контролювати рух в приміщенні та інформувати коритувача звуковою індикацією;
- в) система має ставитись та зніматися з-під охорони;
- г) система має отримувати показники про мікроклімат з метеорологічних датчиків;
- д) система має виводити актуальні показники про мікроклімат на дисплей;
- е) застосування основних принципів робототехніки;
- ж) дотримання технології модульного програмування, бажано фрагменти коду, що мають певне самостійне значення, оформляти у вигляді процедур та функцій, це дозволить покращити розуміння програми і вміння коректно і грамотно користуватись основними прийомами програмування;
- з) реалізація дружнього інтерфейсу: використання вкладок з відповідним набором елементів керування для розуміння, які функції ці елементи реалізують.

Крім того, потрібно розробити супроводжувальну документацію до використання програмно-апаратної системи.

Для реалізації зазначених вимог необхідно вирішити наступні завдання:

1. Розробка схеми підключення компонентів системи.
 - Підключення основного мікроконтролера до датчиків руху, Холла, вологості та температури, CO₂, GSM модуля, модуля ESP8266, RGB світлодіода, звукового сповіщувача, сенсорної кнопки та реле.
2. Програмування алгоритмів обробки даних.
 - Розробка програмного забезпечення для збору даних з датчиків, їх обробки та прийняття рішень про тривожні події.
 - Реалізація логіки для керування виконавчими пристроями та оповіщення користувача.

3. Інтеграція віддаленого керування.

- Розробка програмного інтерфейсу для Wi-Fi модуля ESP8266 з можливістю керування та моніторингу через мобільний додаток або веб-інтерфейс.

4. Реалізація функцій GSM-оповіщення.

- Програмування надсилання SMS у разі спрацювання тривоги.

5. Забезпечення взаємодії між компонентами системи.

- Налаштування зв'язку між основним мікроконтролером та іншими компонентами системи для їх стабільної та надійної роботи.

6. Тестування та оптимізація системи.

- Проведення тестування для перевірки коректності роботи всіх компонентів, виявлення та усунення недоліків.

- Оптимізація програмного коду для забезпечення максимально швидкої реакції системи на події.

Таким чином, основними завданнями є розробка системи охоронної сигналізації з можливістю адаптивного розширення функціоналу та забезпечення високого рівня надійності та безпеки обробки даних.

Функціональні вимоги

1. Виявлення несанкціонованих дій:

- Система повинна своєчасно визначати несанкціоноване відкриття дверей або вікон за допомогою датчика Холла та магніту.

- Система має виявляти рух у контрольованій зоні за допомогою датчика руху та активувати відповідний тривожний сигнал.

- Після активації тривожного сигналу система повинна надсилати сповіщення користувачу через GSM модуль та оновлювати інформацію в інтерфейсі Wi-Fi модуля.

2. Контроль параметрів навколишнього середовища:

- Система повинна здійснювати постійний моніторинг рівня вологості, температури та концентрації CO₂.

- У разі перевищення встановлених граничних значень параметрів середовища система повинна оповістити користувача через дисплей, звукову та світлову індикацію.

3. Передача сповіщень та звітів:

- Система повинна надсилати SMS через GSM модуль у разі тривоги.
- Взаємодія через Wi-Fi з використанням модуля ESP8266 повинна надавати доступ параметрів датчиків та стану системи в режимі реального часу.

4. Віддалене керування та моніторинг:

- Система повинна підтримувати віддалене керування через Wi-Fi: увімкнення/вимкнення режиму охорони, активацію реле, а також налаштування системи.
- Користувач повинен мати можливість переглядати інформацію про поточний стан системи та змінювати її параметри через веб-інтерфейс або мобільний додаток.

5. Візуальна та звукова індикація:

- RGB світлодіод повинен змінювати колір залежно від режиму роботи системи: режим під охороною, режим знято з охорони, режим тривоги.
- Звуковий сповіщувач має забезпечувати звукову індикацію в разі виявлення тривожної події.

6. Керування виконавчими пристроями:

- Система повинна керувати виконавчими пристроями, такими як реле, для включення сирени, аварійного освітлення або блокування дверей у разі активації тривоги.

7. Інтерактивний інтерфейс:

- Дисплей повинен відображати поточні параметри датчиків, статус системи та будь-які попередження чи тривожні сповіщення.

8. Можливість ручного керування:

- Система повинна підтримувати ручне керування за допомогою сенсорної кнопки, яка дозволяє активувати/деактивувати сигналізацію, змінювати режими роботи або здійснювати аварійне вимкнення.

Нефункціональні вимоги

1. Надійність:

- Система повинна забезпечувати безперебійну роботу та стабільну взаємодію між усіма компонентами (датчиками, GSM модулем, ESP8266 тощо).

- Всі компоненти повинні мати резервне живлення на випадок відключення основного електропостачання.

2. Безпека:

- Сповіщення через GSM модуль повинні здійснюватися лише на заздалегідь налаштовані номери телефону.

3. Простота у використанні:

- Інтерфейс користувача повинен бути зрозумілим та інтуїтивно простим для керування.

- Дисплей повинен надавати чітку інформацію про стан системи без зайвих деталей.

4. Сумісність:

- Сумісність з різними версіями GSM модулів та Wi-Fi модулів, що можуть використовуватись у подібних системах.

5. Енергоефективність:

- Система повинна мати низький рівень споживання енергії в режимі очікування, а також забезпечувати енергоефективну роботу датчиків та модулів.

1.4 Висновки

Системи охоронної сигналізації є надзвичайно актуальними в сучасному світі, де безпека є пріоритетом для приватних осіб, підприємств та державних установ. Зростання кількості загроз несанкціонованого доступу стимулює пошук нових, більш ефективних рішень у цій галузі. Комерційні системи охоронної сигналізації, незважаючи на широкий функціонал, мають ряд обмежень, таких як висока вартість, обмежена гнучкість, складність інтеграції з іншими системами та закриті протоколи. Платформа Arduino дозволяє створити більш гнучку, адаптивну та економічно ефективну систему охоронної сигналізації, яка відповідає індивідуальним потребам користувачів. Завданням дослідження є розробка системи охоронної сигналізації на основі платформи Arduino, яка забезпечуватиме контроль доступу, моніторинг параметрів мікроклімату та віддалене керування.

Система повинна бути надійною, безпечною, простою у використанні та мати інтуїтивно зрозумілий інтерфейс. Важливими функціями є виявлення несанкціонованих дій, контроль параметрів навколишнього середовища, передача сповіщень та віддалене керування. Таким чином, розробка власної системи охоронної сигналізації на основі платформи Arduino є перспективним напрямком дослідження, який дозволить створити більш ефективне та адаптивне рішення для забезпечення безпеки об'єктів різного призначення.

2 ПРОЕКТУВАННЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO

2.1 Структурна схема системи охоронної сигналізації на основі платформи Arduino

У сучасному світі безпека житла та об'єктів є одним із головних пріоритетів. Автоматизовані системи охоронної сигналізації, які базуються на мікроконтролерних платформах, дозволяють ефективно забезпечувати контроль та моніторинг навколишнього середовища. Платформа Arduino, завдяки своїй універсальності, простоті програмування та широкій підтримці периферійних пристроїв, є ідеальним вибором для створення таких систем.

У цьому розділі розглянуто загальну структурну схему охоронної сигналізації, що реалізована на основі Arduino. Система включає різноманітні датчики для забезпечення інформування про несанкціоноване втручання в приміщення чи об'єкт охорони, GSM- і WiFi-модулі для передачі даних, елементи індикації й керування, а також моніторингу навколишнього середовища та умов приміщення. Ця схема демонструє інтеграцію апаратних та програмних компонентів для забезпечення функціональності, надійності та зручності у використанні.

На рис. 2.1 зображена загальна структурна схема програмно-апаратної системи охоронної сигналізації на основі платформи Arduino. Яка собою представляє: мікроконтролер, датчик руху, датчик Холлу, сенсорна кнопка, звуковий сповіщувач, GSM модуль, датчики параметрів мікроклімату, дисплей і світловий сповіщувач для виведення інформації, контролер з WiFi модулем, 2 реле, сервер, мобільний додаток.

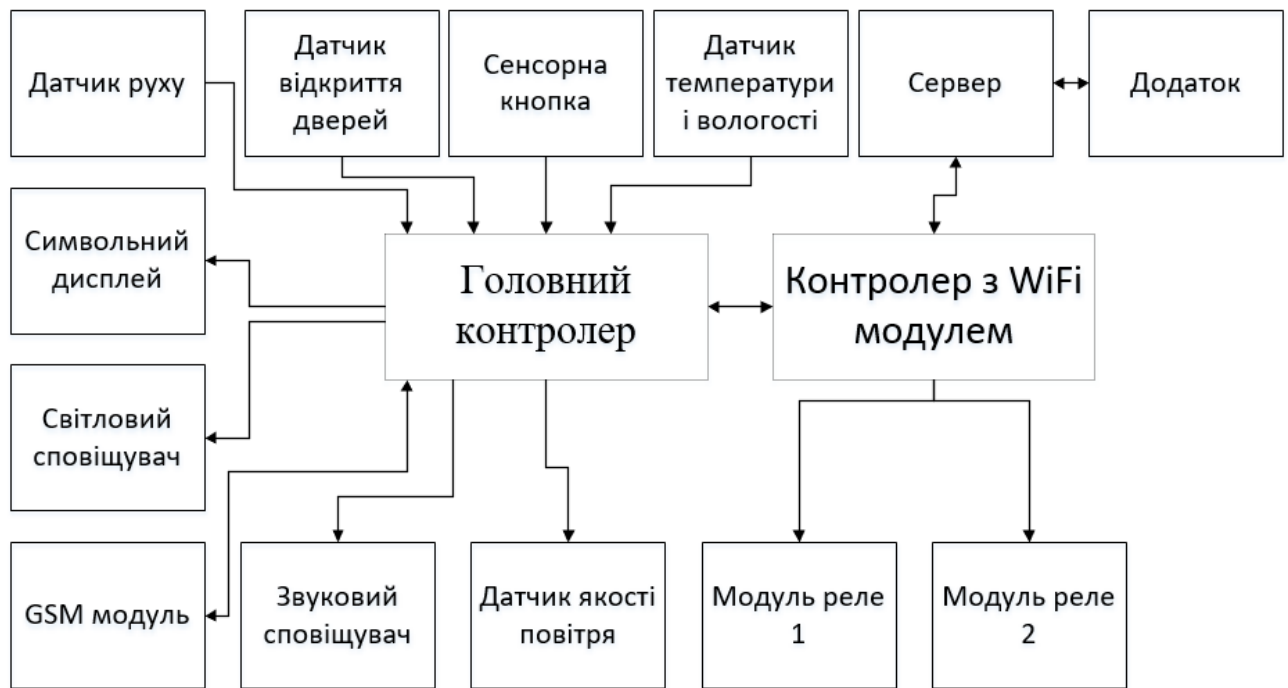


Рисунок 2.1 – Загальна структурна схема системи охоронної сигналізації на основі платформи Arduino

Загальна структурна схема системи охоронної сигналізації складається з таких основних компонентів:

- Головний контролер - здійснює обробку сигналів від датчиків, управління периферійними пристроями та обмін інформацією з іншими модулями. Він отримує дані від датчиків і формує відповідні сигнали для звукових, світлових сповіщувачів та дисплеїв.

- Датчик руху - виявляє переміщення у зоні дії та передає сигнал до мікроконтролера для подальшої обробки.

- Датчик відкриття дверей(датчик Холла) - контролює стан дверей (відчинені/зачинені) та передає відповідну інформацію.

- Датчик температури та вологості - зчитує мікрокліматичні параметри (температуру і вологість) у приміщенні.

- Датчик якості повітря - моніторить концентрацію шкідливих речовин та рівень CO₂ у повітрі.

- Сенсорна кнопка - забезпечує можливість взаємодії користувача із системою для зміни стану системи.

- Символьний дисплей - використовується для виведення даних про стан системи, наприклад, стан охорони, стан дверей, значень температури, вологості чи якості повітря.

- Світловий сповіщувач - світлодіодний індикатор, що сигналізує про стан системи або тривогу.

- Звуковий сповіщувач - видає звуковий сигнал у разі спрацювання датчика чи тривоги.

- GSM-модуль - забезпечує надсилання повідомлень (SMS) або дзвінків для інформування користувача про стан системи чи спрацювання датчиків.

- Контролер із WiFi-модулем – ще один мікроконтролер який дозволяє системі підключатися до мережі Інтернет для взаємодії із сервером, який у свою чергу взаємодіє з мобільним додатком.

- Реле слугують для керування виконавчими пристроями, наприклад, електронними замками чи системами вентиляції. У схемі передбачено два незалежні реле для розширення функціональності.

Принцип роботи системи охоронної сигналізації полягає у зборі, обробці та передачі даних для забезпечення моніторингу та оперативного інформування користувача про стан об'єкта. Датчик відчинення дверей та датчик руху передають інформацію до мікроконтролера, мікроконтролер спілкується з GSM модулем для обміну інформації по SIM. Датчики температури, вологості та якості повітря передають значення відповідних фізичних величин на мікроконтролер. Символьний дисплей та світлодіод слугує для інформування користувача про стан системи та потрібною інформацією. Для прикладу, на дисплей виводиться значення температури в градусах Цельсія, яке передали датчики мікроклімату. Контролер з WiFi модуль, який під'єднується до наявної точки доступу, через мережу інтернет з'єднується з сервером, який у свою чергу, з'єднує мобільний додаток та систему.

2.2 Розробка монтажної схеми охоронної сигналізації на основі платформи Arduino

Для досягнення цілі важливо правильно вибрати і інтегрувати всі компоненти системи. Від того, як здійснено підключення датчиків, модулів та інших елементів, залежить стабільність роботи системи, її здатність правильно обробляти та передавати інформацію.

Розробка монтажної схеми є одним з основних етапів проектування системи охоронної сигналізації на основі платформи Arduino. Монтажна схема відображає підключення всіх компонентів до основного мікроконтролера та взаємодію між ними. Правильно спроектована схема забезпечує стабільність роботи системи, коректну передачу сигналів між датчиками та мікроконтролером, а також мінімізує можливість виникнення збоїв у роботі обладнання.

Монтажна схема системи має мати вигляд, який показаний на рис. 2.2 .

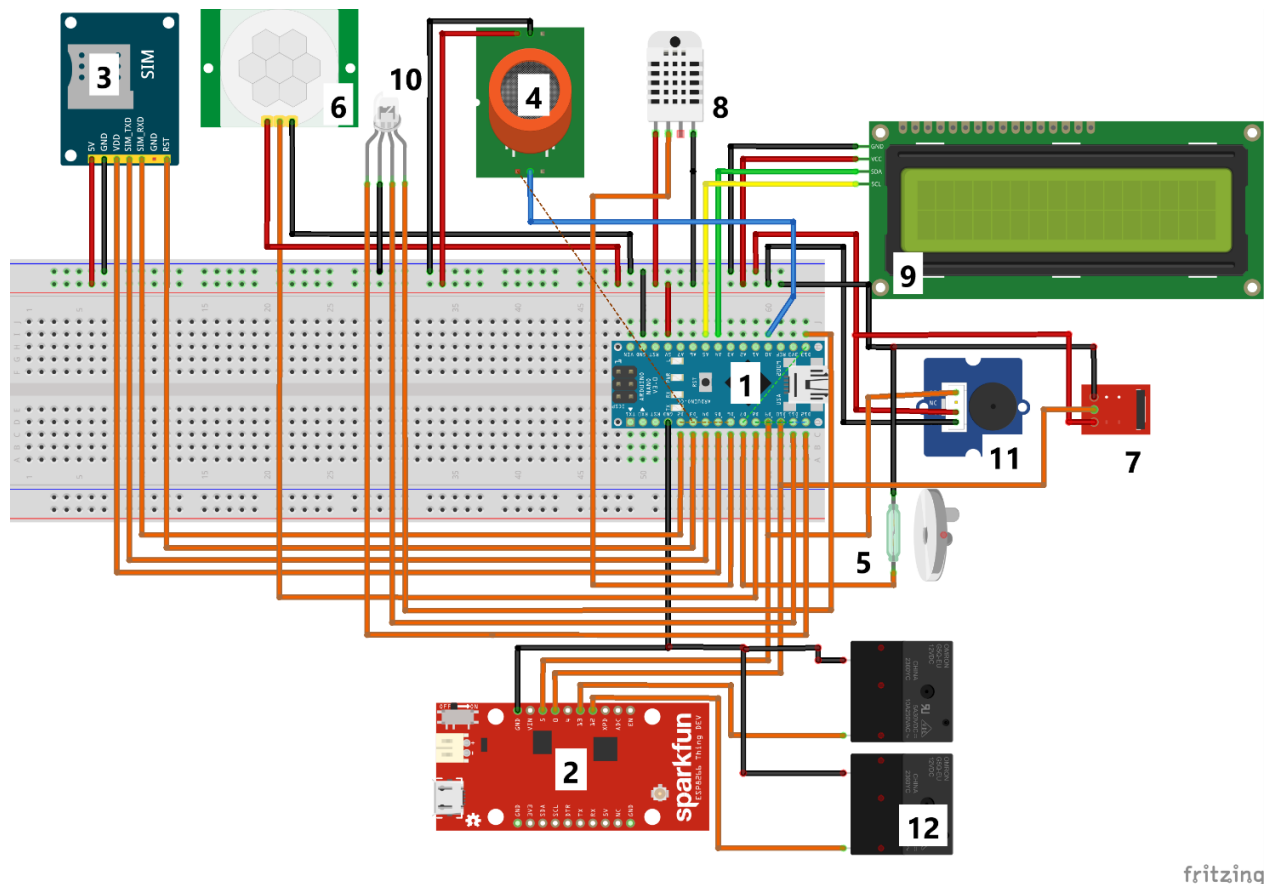


Рисунок 2.2– Монтажна схема системи охоронної сигналізації на основі платформи Arduino

На рис. 2.2 прийнято такі позначення: 1 - головний контролер, 2 - контролера з WiFi модулем, 3 - GSM модуль, 4 - модуль для вимірювання якості повітря, 5 - датчика для контролю відкриття дверей, 6 - датчик руху, 7 – кнопка, 8 - модуль для зчитування параметрів мікроклімату, 9 - модуль для відображення інформації, 10 – світлового сповіщувача, 11 – звукового сповіщувача, 12 – модуля для керування автоматикою (реле).

В якості головного контролера пропонується обрати - Arduino Nano це функціональний аналог Arduino Uno, але розміщений на мініатюрній платі тому була вибрана цей мікроконтролер для збереження компактності. Arduino Nano - є компактним, доступним і має достатню кількість входів і виходів для підключення різних датчиків та компонентів системи. Це добре перевірена платформа для швидкої розробки і тестування. Характеристики: Мікроконтролер ATmega328P, підтримує 5V напругу, має 14 цифрових входів/виходів та 8 аналогових входів [15].

В якості контролера з WiFi модулем пропонується обрати - ESP8266 має менше енергоспоживання в порівнянні з ESP32, що робить його кращим вибором для проектів з обмеженими джерелами енергії. ESP8266 вибір цього мікроконтролера обумовлений необхідністю наявності WiFi модуля для підключення до інтернету та зв'язку з сервером через мережу. ESP8266 має достатню обчислювальну потужність і великий потенціал для роботи з сервісами, такими як мобільний додаток. Характеристики: Мікроконтролер з підтримкою Wi-Fi, працює при напрузі 3.3V, має достатньо пінів для підключення зовнішніх датчиків і модулів [16].

В якості GSM модуля пропонується обрати - SIM800, для забезпечення зв'язку через GSM-мережу. SIM800 є більш доступним за ціною, ніж деякі інші модулі, такі як SIM808 чи SIM5320. SIM800 має відносно низьке енергоспоживання. Модуль підтримує передачу даних через SMS та голосові дзвінки, що дозволяє оперативно сповістити користувача про тривогу або інші події. Характеристики: Підтримка GSM (2G) мережі, може працювати з різними типами SIM-карт, підтримка передачі SMS та голосових викликів [17].

В якості модуля для вимірювання якості повітря пропонується обрати - MQ135, є універсальним сенсором, який може виявляти декілька газів, зокрема

аміак (NH_3), бензол (C_6H_6), вуглекислий газ (CO_2), сірководень (H_2S), оксиди азоту (NO_x), а також леткі органічні сполуки (VOC). Це дозволяє застосовувати його для моніторингу загального стану повітря в приміщеннях, де можуть бути присутні різні токсичні та шкідливі гази. Інші аналоги, такі як MQ2, MQ3, чи MQ7, зазвичай обмежені одним або кількома газами (наприклад, тільки CO , або лише метан), що робить їх менш універсальними для загального моніторингу якості повітря. Характеристики: Датчик аналізує концентрацію газів, таких як аміак, CO_2 , бензол і інші леткі органічні сполуки [18].

В якості датчика для контролю відкриття дверей пропонується обрати - Датчик Холла, використовується для виявлення магнітного поля, що дозволяє реалізувати функцію контролю дверей або вікон. Він є надійним і не має рухомих частин. Характеристики: працює за принципом вимірювання магнітного поля [19].

В якості датчика руху пропонується обрати - HC-SR501, цей датчик пасивного інфрачервоного (PIR) типу використовується для виявлення руху людей. AM312 Mini PIR Sensor це також датчик руху PIR, але менший за розміром і з меншою чутливістю. Він чудово підходить для компактних пристроїв, але має обмежений кут огляду та налаштування у порівнянні з HC-SR501, AM312 менш гнучкий у налаштуванні. HC-SR501 має два регулятори для налаштування чутливості та тривалості затримки сигналу. Він ідеально підходить для безпеки, оскільки дозволяє виявити рух в приміщенні. Характеристики: Датчик має високу чутливість до руху, працює при напрузі 5V [20].

В якості кнопки пропонується обрати - TTP223 (ємнісна сенсорна кнопка) вибрана для створення інтерфейсу взаємодії з користувачем. Вона є безконтактною, що забезпечує додаткову зручність і надійність. Характеристики: чутливість до дотику без фізичних кнопок, що збільшує довговічність і надійність [21].

В якості модуля для зчитування параметрів мікроклімату пропонується обрати - DHT22 (датчик температури та вологості) обрано для моніторингу температури та вологості в приміщенні. DHT22 має точність вимірювання температури $\pm 0.5^\circ\text{C}$, що значно краще за $\pm 2^\circ\text{C}$ у DHT11. Точність вологості також є кращою у DHT22 — близько $\pm 2\text{-}5\%$ RH проти $\pm 5\%$ RH у DHT11. DHT22 може вимірювати

температуру в діапазоні від -40°C до $+80^{\circ}\text{C}$, тоді як DHT11 обмежений від 0°C до $+50^{\circ}\text{C}$. Діапазон вимірювання вологості для DHT22 становить від 0% до 100% RH, у той час як для DHT11 цей діапазон лише від 20% до 90%. Це важливі параметри для забезпечення комфорту в приміщенні. Характеристики: Вимірює температуру в діапазоні від -40 до 80°C і вологість від 0 до 100%, працює при 5V [22].

В якості модуля для відображення інформації пропонується обрати - LCD 2004 I2C, для виведення інформації на LCD дисплей з I2C інтерфейсом. Завдяки інтерфейсу I2C потрібно лише два провідники (SDA та SCL) для передачі даних, що значно спрощує підключення і звільняє порти мікроконтролера для інших задач. LCD 2004 має 20 символів у 4 рядках, що дозволяє виводити більше інформації, порівняно з LCD 1602 (16 символів на 2 рядки). Характеристики: Дисплей з 20 символами на 4 рядки, підключення через I2C для зручності і економії пінів мікроконтролера [23].

В якості світлового сповіщувача пропонується обрати - RGB-світлодіод, використовується для індикації стану системи або для візуального попередження про тривогу або інші події. Характеристики: Дозволяє комбінувати червоний, зелений та синій кольори для виведення різних індикаторів [24].

В якості звукового сповіщувача пропонується обрати - модуль з динаміком (Low-level), вибраний для звукового сповіщення про тривогу або інші загрози, такі як пожежа. Це важливий елемент для системи безпеки. Характеристики: Підтримує різні звукові сигнали, працює при 5V напрузі [25].

В якості модуля для керування автоматикою пропонується обрати - 1-канальний реле модуль 5V, необхідний для управління зовнішніми пристроями, такими як сирена або сигналізація, в разі спрацьовування системи. 1-канальний реле модуль легко підключати до мікроконтролера, оскільки він потребує мінімальної кількості контактів та живлення 5V, яке можна отримати безпосередньо з контролера. Характеристики: Підтримка 5V напруги, може керувати навантаженнями до 10A, що достатньо для роботи більшості пристроїв [26].

2.3 Розробка UML-діаграм функціонування системи охоронної сигналізації

UML – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Вона є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація [27].

Одне із завдань UML – служити засобом комунікації всередині команди та при спілкуванні з замовником.

Розглянемо можливі варіанти використання діаграм:

- Проектування. UML-діаграми стануть у пригоді при моделюванні архітектури великих проектів, в якій можна зібрати як великі, так і дрібніші деталі і намалювати каркас (схему) програми. По ньому пізніше буде будуватись код.
- Реверс-інжиніринг – створення UML-моделі з існуючого коду додатку, зворотна побудова. Може застосовуватися, наприклад, на проектах підтримки, де є написаний код, але документація неповна або відсутня.
- З моделей можна витягувати текстову інформацію і генерувати відносно читабельні тексти – документувати. Текст і графіка будуть доповнювати один одного.

В мові UML є 12 типів діаграм:

- 4 типи діаграм представляють статичну структуру додатку;
- 5 типів представляють поведінкові аспекти системи;
- 3 представляють фізичні аспекти функціонування системи (діаграми реалізації).

Для реалізацій розробки обрано 2 UML діаграми:

- діаграма станів;

- діаграма комунікації [28].

На рис. 2.3 зображена UML діаграма комунікації програмно-апаратної системи.

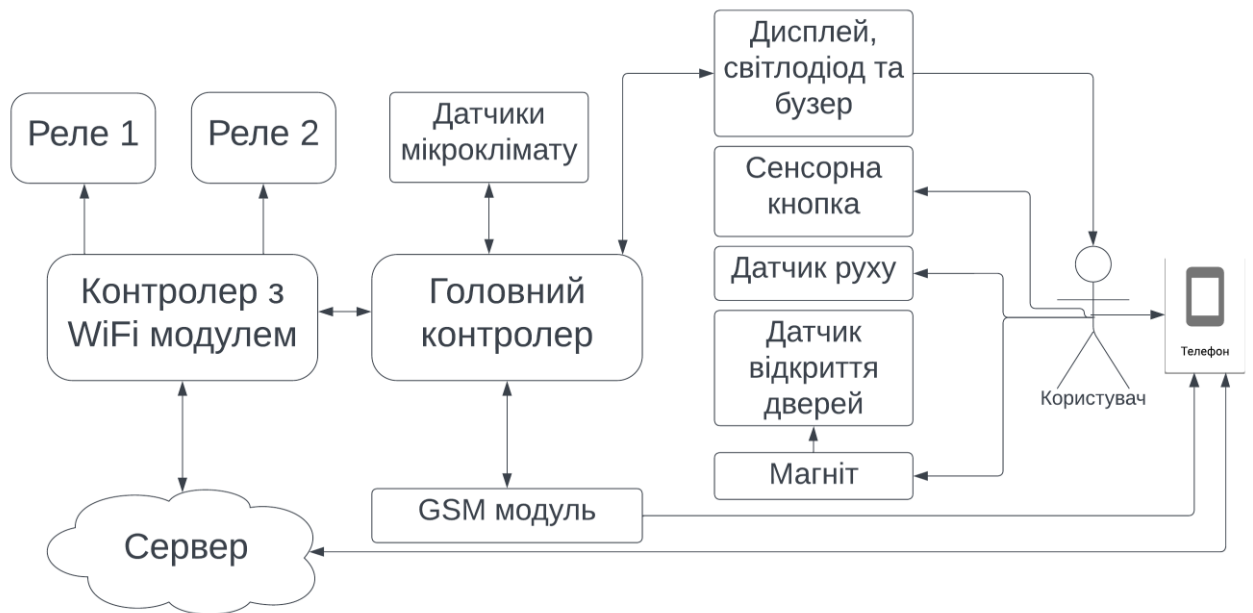


Рисунок 2.3 – UML діаграма комунікації системи охоронної сигналізації на платформі Arduino

На UML діаграмі комунікації зображено роботу системи охоронної сигналізації. Діаграма відображає взаємодію між основними компонентами системи:

Мікроконтролер – головний компонент, який керує системою. Він отримує дані від датчиків. Контролер з WiFi модулем – забезпечує зв'язок з сервером та передає дані від мікроконтролера до сервера, а також керує двома реле. Реле – реле, які керують зовнішніми пристроями на основі отриманих команд. Сервер – отримує дані з контролера через WiFi модуль і може передавати команди для управління системою. GSM модуль – забезпечує передачу повідомлень користувачу у разі виявлення події або тривоги. Датчики мікроклімату – передають інформацію про кліматичні умови (температура, вологість, CO₂) на мікроконтролер. Дисплей, світлодіод та звуковий індикатор – слугують для візуального та звукового інформування користувача про стан системи. Сенсорна кнопка – надає можливість взаємодії користувача із системою. Датчик руху – відслідковує рух і сигналізує про

рух на мікроконтролер. Датчик відкриття дверей – використовуються для виявлення відкриття дверей або вікон. Якщо магніт віддаляється від датчика, це сигналізує про тривогу. Користувач (Телефон) – отримує повідомлення про стан системи або сповіщення про тривогу.

На рис. 2.4 зображена UML діаграма станів програмно-апаратного системи охоронної сигналізації на платформі Arduino.

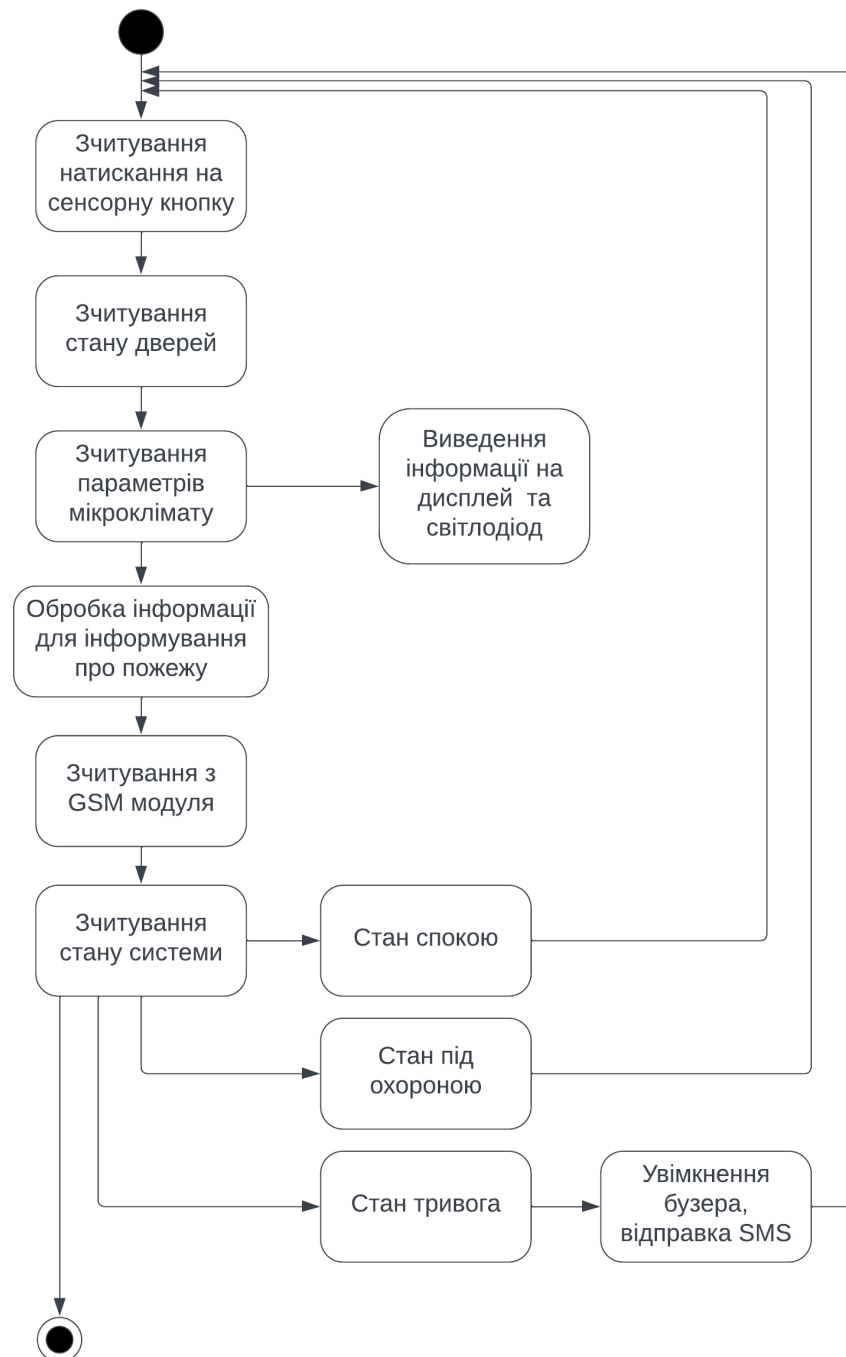


Рисунок 2.4 – UML діаграма станів системи охоронної сигналізації на платформі Arduino

Спочатку зчитується стан сенсорної кнопки - система перевіряє натискання сенсорної кнопки. Це викликає зміну стану, наприклад, перевести систему в режим під охороною або режим спокою. Читання статусу дверей -система перевіряє, відкриті чи закриті двері. Це впливає на дозвіл зміни стану системи. Зчитування параметрів мікроклімату - система зчитує сигнали від датчиків мікроклімату (температура, вологість, рівень газу). Дані виводяться на дисплеї, що дозволяє контролювати умови мікросередовища. Відображення інформації на екрані та світлодіоді - система відображає на екрані мікрокліматичні умови та стан системи. Обробка інформації для пожежної сигналізації - якщо кліматичні параметри показують задимлення або високу температуру, стан системи тривога.

Стан спокою - система знаходиться в режимі спокою і не реагує на події такі як рух, відкриття дверей, але стежить за мікрокліматом.

Стан під охороною - система знаходиться в режимі під охороною і реагує на події такі як рух, відкриття дверей. Якщо є порушення, рух чи двері відкриті, то система переводиться в стан тривога.

Стан під охороною - вмикається звуковий сигнал тривоги, змінюється інформація на дисплеї, змінюється колір світлового сповіщувача.

2.4 Розробка алгоритмів функціонування програмно-апаратної системи охоронної сигналізації на основі платформи Arduino

Для розробки алгоритма роботи програмно-апаратної системи охоронної сигналізації обрано схематичний спосіб опису алгоритму. Такий спосіб опису передбачає опис алгоритму за допомогою структурних блоків. Великою перевагою є те, що потрібно лише один раз розробити схему та подати її програмісту.

Алгоритм роботи ESP8266 подано на рис. 2.5.

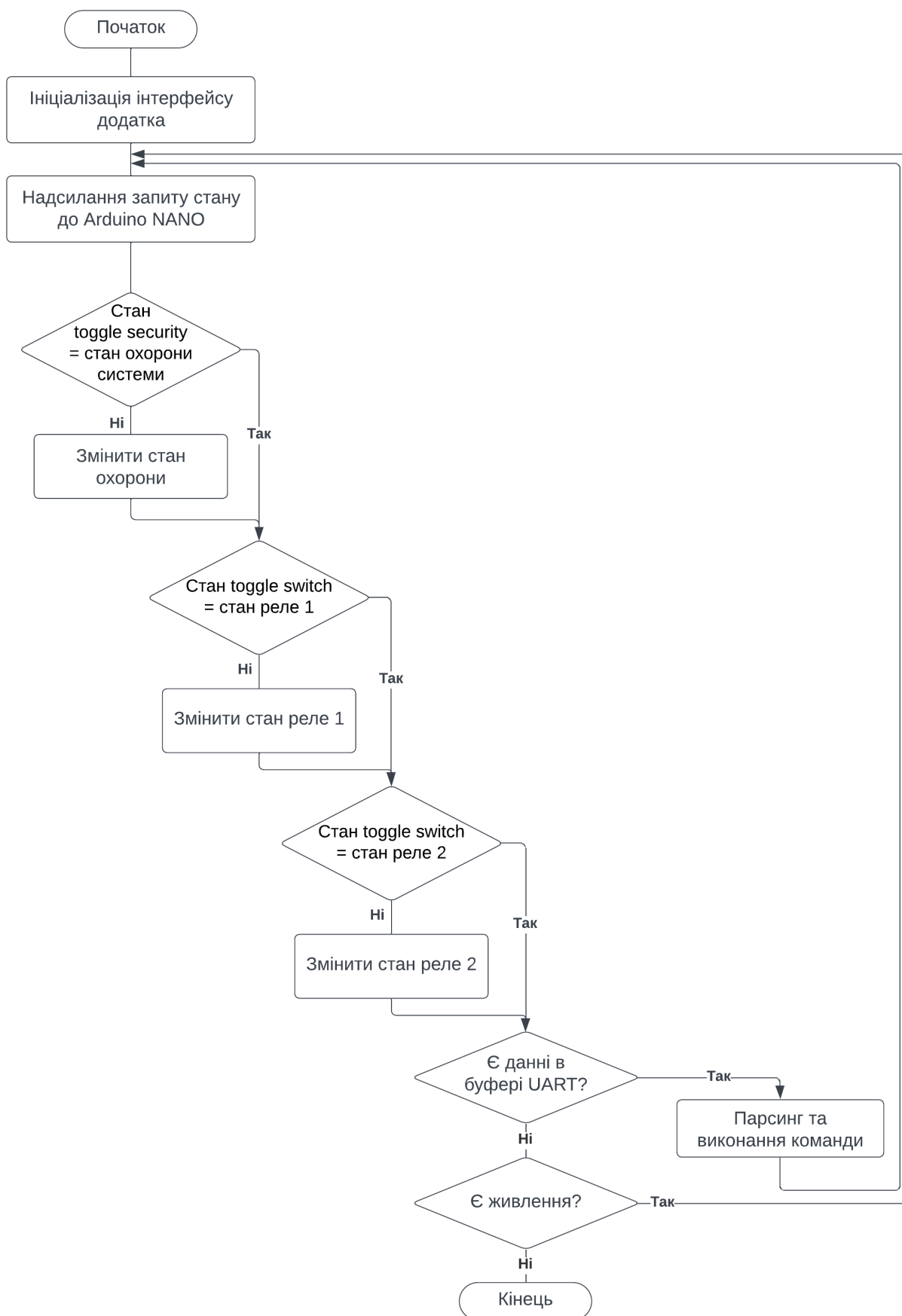


Рисунок 2.5 – Схема алгоритму роботи контролера з WiFi

Алгоритм роботи контролера з WiFi починається з ініціалізації інтерфейсу додатка. Далі надсилається запит до головного мікроконтролера для отримання актуального стану системи охорони. Перевірка стану охорони полягає в тому, якщо стан toggle security (в інтерфейсі) не відповідає стану охоронної системи, то - виконати зміну стану охорони відповідно до стану toggle security. Перевірка стану реле 1 - якщо стан toggle switch (в інтерфейсі) не відповідає стану реле 1, то - змінити стан реле 1 відповідно до стану toggle switch. Перевірка стану реле 2 - якщо стан toggle switch (в інтерфейсі) не відповідає стану реле 2, то змінити стан реле 2 відповідно до стану toggle switch. Перевірка наявності даних у буфері UART, якщо в UART-буфері є дані, то виконати парсинг даних та обробку отриманих команд. Далі відбувається перевірка живлення. Якщо живлення відсутнє, завершити роботу. Якщо живлення є, повернутися до початкових дій для циклічного виконання перевірок.

Алгоритм роботи головного контролера системи охорони починається після подачі живлення на контролер, далі зчитується чи є натискання на кнопку активації. Спочатку перевіряється стан системи охорони - якщо вона вже під охороною, її можна зняти, а якщо не під охороною, то вона ставиться під охорону. Далі перевіряється стан дверей - якщо двері відкриті, зчитуються показники мікроклімату та виводяться на дисплей. У разі, якщо температура і рівень CO₂ перевищують допустимі значення, система автоматично відправляє SMS про пожежу. Потім перевіряється наявність SMS в буфері GSM та наявність даних у буфері UART і виконання відповідних команд. Залежно від стану системи (під охороною чи знята з охорони), змінюється колір світлодіода, відображається відповідне повідомлення на дисплеї, а також вимикається контроль датчиків руху та геркону. Якщо виникає тривога, активується червоний колір світлодіода, а на дисплеї з'являється повідомлення про "Alarm". В останньому кроці перевіряється наявність живлення, після чого алгоритм або продовжує роботу, або завершується.

Алгоритм роботи головного контролера подано на рис. 2.6.

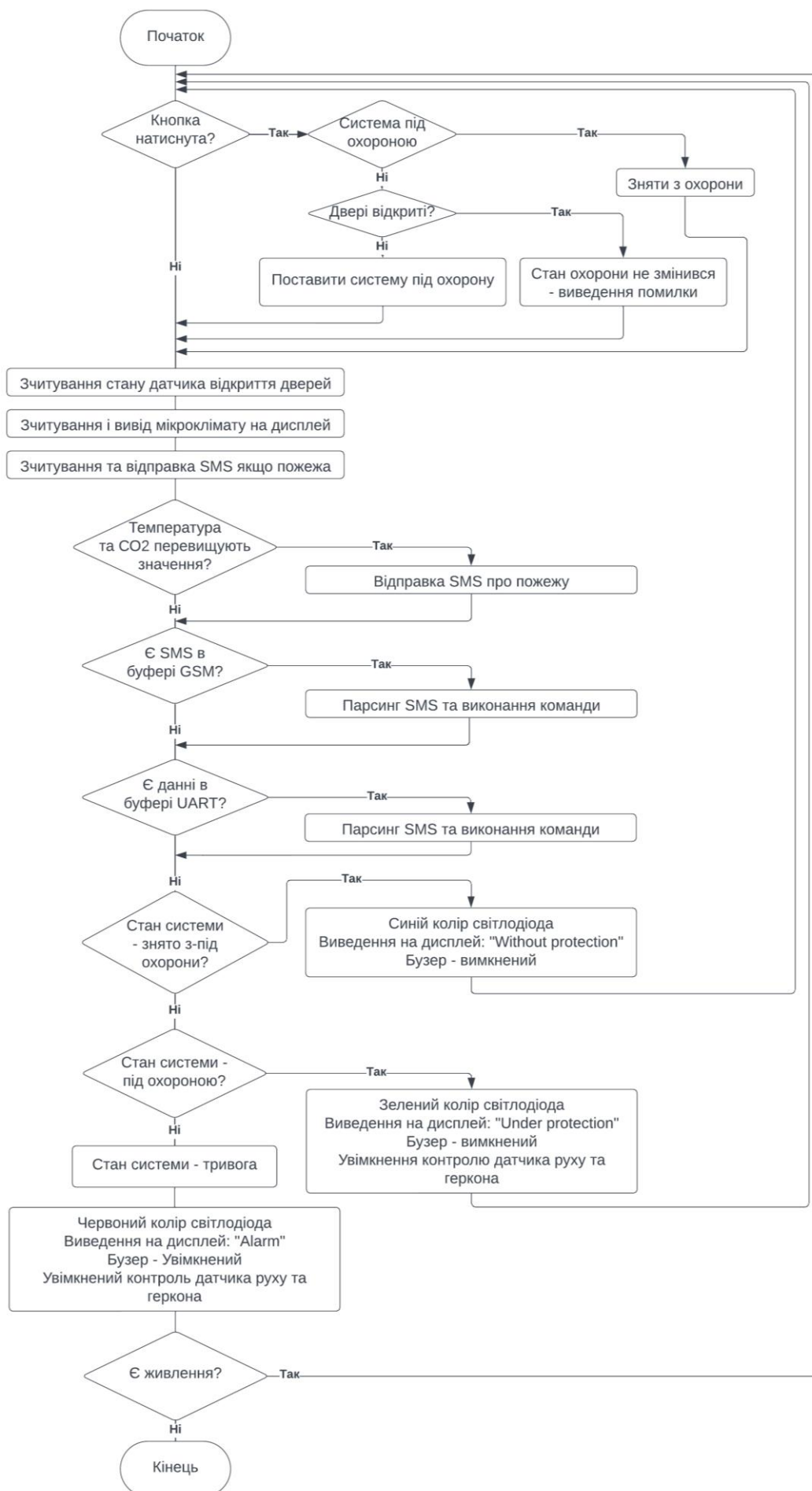


Рисунок 2.6 – Схема алгоритму роботи головного контролера

2.5 Аналіз потенційних загроз та оцінювання ефективності системи охоронної сигналізації

Житлове приміщення піддається багатьом потенційним загрозам, які можуть бути спричинені діями зловмисників, технічними збоями, кіберзагрозами, стихійними лихами. Для забезпечення безпеки необхідно використовувати охоронні системи, контролювати якість повітря та вологість, застосовувати сучасні системи захисту. Інтеграція системи «розумний дім» та автоматизація процесів моніторингу дозволяють оперативно реагувати на загрози та мінімізувати можливі збитки.

Потенційні загрози приміщенню: несанкціонований доступ, злом дверей, вікон або інших конструкцій, проникнення, крадіжка майна, пошкодження приміщення, саботаж магнітним полем, глушіння GSM каналу, глушіння WiFi каналу, загроза життю мешканців, підпал, вандалізм, несправність електроприладів, несправність газових приладів, витік природного газу, вибух, пожежа, протікання труб, прориви системи водопостачання, затоплення, несправність вентиляції, знищення системи фізичним впливом.

Потенційні загрози несанкціонований доступ, злом дверей, вікон або інших конструкцій, проникнення, крадіжка майна, пошкодження приміщення, загроза життю мешканців, вандалізм покриваються системою за рахунок використання датчика відчинення дверей та датчика руху. Загроза саботаж магнітним полем неможливе за рахунок використання датчика Холла замість звичайного геркона. Загрози підпал, пожежа, несправність газових приладів, несправність вентиляції, витік природного газу покриваються системою за рахунок використання датчика температури та датчика якості повітря, які сигналізують про перевищення допустимих параметрів та виявлення диму або інших продуктів горіння. Загрози несправність електроприладів, протікання труб, прориви системи водопостачання, затоплення покриваються системою за рахунок використання реле якими можна зняти живлення в електроприладів, чи використовувати електромагнітний клапан для перекриття водопостачання.

Загроза «знищення системи фізичним впливом» впливає з того, що будь-який фізичний об'єкт, навіть захищений корпусом, можна зруйнувати механічно. Захист від цього потребує додаткових фізичних бар'єрів або прихованого монтажу, щоб не дати безпосередній доступ зловмисника до системи. Також система не покриває таку загрозу як «вибух».

Загроза «глушіння GSM каналу» та «глушіння WiFi каналу» впливає з того, що GSM модуль не може підключитися до базової станції оператора зв'язку, оскільки всі доступні частоти зайняті "шумом". Система не може "пробити" цей шум, адже передача сигналу базується на тих самих частотах, що і глушіння. GSM-канали працюють у жорстко визначених частотних діапазонах (900/1800/2100 MHz). Якщо всі ці діапазони заблоковані, пристрої не можуть "перемкнутися" на інші частоти, оскільки вони не передбачені для GSM-модулів. Джамери створюють радіоперешкоди в частотних діапазонах Wi-Fi (2.4 GHz та 5 GHz). Wi-Fi передача сигналу блокується так само, як і для GSM, шляхом "заливання" шуму у відповідному діапазоні. Але ці загрози не блокують роботу системи за рахунок одночасного використання 2-х каналів зв'язку GSM та Wi-Fi, цей спосіб найефективніших для підвищення стійкості системи охоронної сигналізації до глушіння, використання подвійного каналу зв'язку (GSM + Wi-Fi). Цей підхід суттєво ускладнює дії зловмисників, оскільки для повного відключення сигналізації їм потрібно буде одночасно заглушити обидва канали зв'язку, що потребує значно більш складного обладнання та ресурсів.

На рис 2.7 показано перелік потенційних загроз, які система охоронної сигналізації покриває.

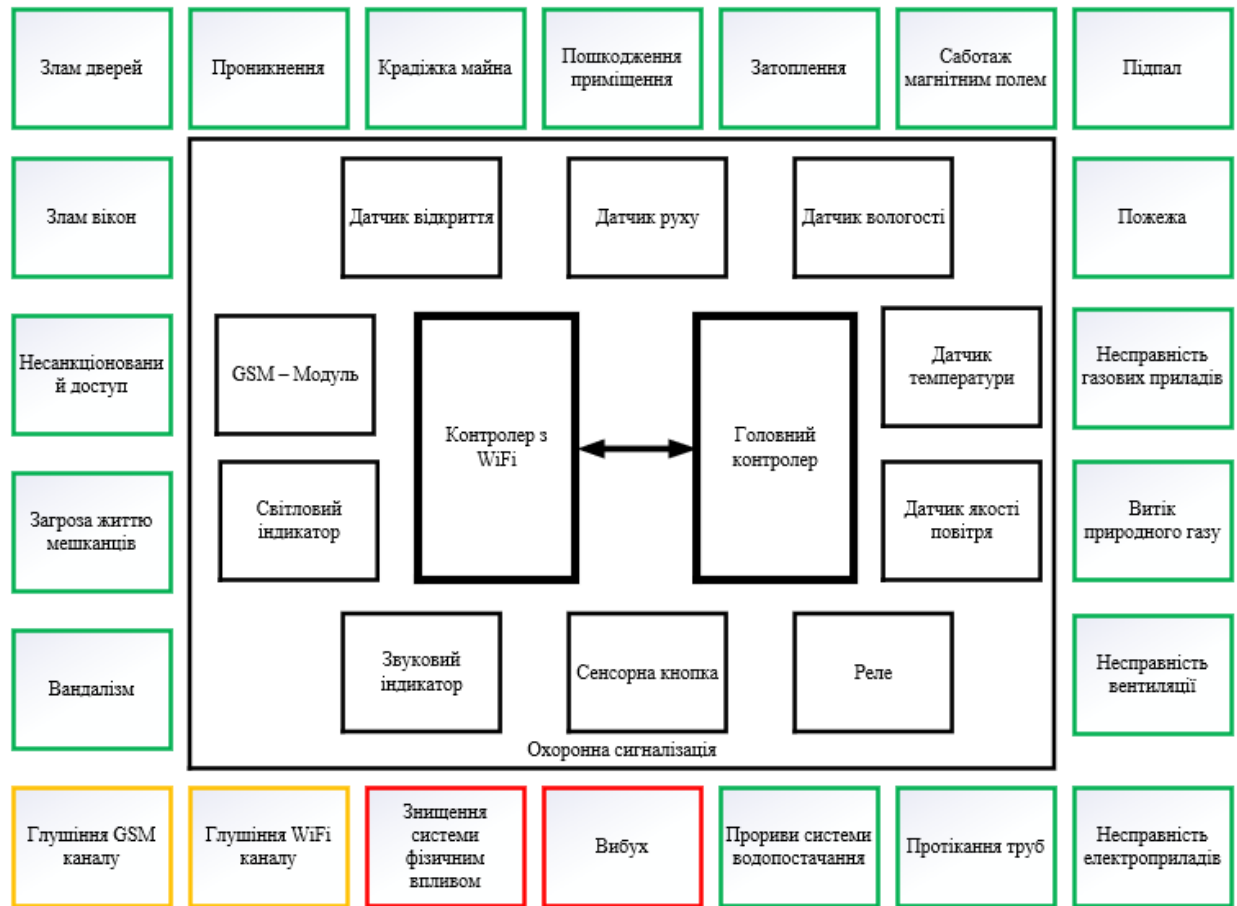


Рисунок 2.7 – Перелік потенційних загроз, які система охоронної сигналізації покриває

Для оцінки системи можна розрахувати коефіцієнт ефективності захисту:

$$K_{\text{protect}} = \frac{N_{OC}}{N_{All}} \cdot 100\% \quad (2.1)$$

де:

- N_{OC} — кількість загроз, які система покриває;
- N_{All} — загальна кількість потенційних загроз.

Підставляємо дані:

- Загроз, які система покриває = 18
- Загальна кількість потенційних загроз = 22

$$K_{\text{protect}} = \frac{18}{22} = 0.82$$

Таким чином, коефіцієнт ефективності захисту системи охоронної сигналізації становить 0.82, що свідчить про високий рівень забезпечення захисту від потенційних загроз.

2.6 Теоретико-множинна модель системи охоронної сигналізації

Запропоновану систему охоронної сигналізації можна представити у вигляді такої моделі:

$$OC = \{ C, I, O, S, E \},$$

де:

C — множина компонентів системи охоронної сигналізації,

I — множина вхідних сигналів,

O — множина вихідних сигналів,

S — множина станів системи,

E — множина подій, які ініціюють зміну стану системи.

Модель системи охоронної сигналізації на основі платформи Ардуіно, можна визначити такою множиною компонентів:

$$C = \{ C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12} \},$$

де:

C₁ — головний контролер (Arduino Nano),

C₂ — контролер з Wi-Fi (ESP8266),

C₃ — GSM модуль (SIM800),

C₄ — датчик якості повітря (MQ135),

C₅ — датчик контролю відкриття дверей (датчик Холла),

C₆ — датчик руху (HC-SR501),

C₇ — сенсорна кнопка (TTP223),

C₈ — датчик мікроклімату (DHT22),

C₉ — модуль відображення інформації (LCD 2004 I2C),

C₁₀ — RGB-світлодіод (світловий сповіщувач),

C₁₁ — звуковий сповіщувач (бuzzer),

C₁₂ — реле (1-канальний модуль 5V).

Множина вхідних сигналів системи:

$$I = \{ I_1, I_2, I_3, I_4, I_5, I_6, I_7 \},$$

де:

I_1 – сигнал від датчика Холла (відкриття/закриття дверей),

I_2 – сигнал від датчика руху (виявлення руху),

I_3 – стан сенсорної кнопки (натиснута/не натиснута),

I_4 – показники температури від DHT22,

I_5 – показники вологості від DHT22,

I_6 – показники газів (CO_2) від MQ135,

I_7 – команди від користувача через UART або GSM.

Множина вихідних сигналів системи:

$$O = \{ O_1, O_2, O_3, O_4, O_5, O_6 \},$$

де:

O_1 – стан реле (ввімкнено/вимкнено),

O_2 – колір і стан RGB-світлодіода (індикація охорони, тривоги),

O_3 – сигнал звукового сповіщувача (звук тривоги),

O_4 – повідомлення на дисплеї (LCD 2004),

O_5 – повідомлення, що надсилаються через GSM (SMS),

O_6 – відправка даних через Wi-Fi (на сервер або мобільний додаток).

Визначення функціональних множин. Множина станів системи охоронної сигналізації:

$$S = \{ S_1, S_2, S_3, S_4 \},$$

де:

S_1 – Система під охороною (активна охорона),

S_2 – Система знята з охорони (неактивна охорона),

S_3 – Стан тривоги (активована тривога),

S_4 – Система в режимі аварійного вимкнення (відсутність живлення).

Множина подій системи:

$$E = \{ E_1, E_2, E_3, E_4, E_5, E_6, E_7 \}$$

де:

E_1 – натискання сенсорної кнопки (постановка/зняття з-під охорони),

E_2 – порушення датчика Холла (відкриття дверей),

E_3 – активація PIR датчика (виявлено рух),

E_4 – виявлення небезпечних газів (CO_2),

E_5 – команда через UART (керування системою),

E_6 – команда через GSM (керування системою),

E_7 – відсутність живлення (перехід у режим аварійного вимкнення).

Функціональні зв'язки системи. Вхідні дані $\rightarrow$ Система $\rightarrow$ Вихідні сигнали:

$$f: I \times S \rightarrow O$$

Функція f визначає, як на основі поточного стану системи S та вхідних сигналів I формуються вихідні сигнали O .

Функція переходу між станами:

$$\delta: S \times E \rightarrow S,$$

Де δ — функція переходу між станами. Залежно від подій E , система змінює свій стан S . Розглянемо

Умови переходу початкового стану S_2 (система знята з охорони):

- E_1 (натискання кнопки) $\rightarrow S_1$ (система під охороною).
- E_5 (команда UART) $\rightarrow S_1$ (система під охороною).
- E_6 (команда GSM) $\rightarrow S_1$ (система під охороною).
- E_7 (відсутність живлення) $\rightarrow S_4$ (вимкнення).

Логічні умови функціонування системи можна представити у вигляді таких правил:

1. Якщо S_1 (під охороною) і I_1 (двері відкрито), то S_3 (тривога) та O_2 (червоний колір RGB) і O_3 (звук тривоги) і O_4 (повідомлення на дисплеї) і O_5 (повідомлення, що надсилаються через GSM) і O_6 (відправка даних через Wi-Fi).
2. Якщо S_1 (під охороною) і $I_2 = 1$ (виявлення руху), то S_3 (тривога) та O_2 (червоний колір RGB) і O_3 (звук тривоги) і O_4 (повідомлення на дисплеї) і O_5 (повідомлення, що надсилаються через GSM) і O_6 (відправка даних через Wi-Fi).
3. Якщо I_6 (CO_2 високий) і I_4 (висока температура), то S_3 (тривога) та O_2 (червоний колір RGB) і O_3 (звук тривоги) і O_4 (повідомлення на дисплеї) і O_5 (повідомлення, що надсилаються через GSM) і O_6 (відправка даних через Wi-Fi).
4. Якщо E_1 (натискання кнопки) і S_2 (система знята з охорони) і I_1 (двері відкрито), то S_2 (система знята з охорони) та O_4 (повідомлення на дисплеї).

5. Якщо E_1 (натискання кнопки) і S_2 (система знята з охорони) і I_1 (двері закриті), то S_1 (під охороною) та O_4 (повідомлення на дисплеї).
6. Якщо E_5 (команда UART), то O_1 (зміна стану реле).

2.7 Висновки

У цьому розділі було представлено детальну структурну схему системи, яка включає всі необхідні компоненти для забезпечення функціональності системи. Кожен компонент був детально описаний та обґрунтовано його вибір. Наведена монтажна схема візуалізує підключення всіх компонентів та забезпечує чітке розуміння реалізації системи. Були розроблені діаграми комунікації та станів, які наочно демонструють взаємодію між компонентами системи та логіку її роботи. Представлені алгоритми роботи головного контролера та контролера з Wi-Fi модулем, які описують послідовність виконання операцій та прийняття рішень. Було проведено аналіз потенційних загроз та розроблена математична модель системи, яка дозволяє формально описати її структуру та функціонування.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ПЛАТФОРМІ ARDUINO

3.1 Обґрунтування вибору платформ для розробки системи охоронної сигналізації

Для розробки програмної частини системи використовувалося програмне середовище Arduino IDE.

Arduino – це ефективний засіб розробки програмованих електронних пристроїв, які, на відміну від персональних комп'ютерів, орієнтовані на тісну взаємодію з навколишнім світом. Arduino – це відкрита програмована апаратна платформа для роботи з різними фізичними об'єктами і являє собою просту плату з мікроконтролером, а також спеціальне середовище розробки для написання програмного забезпечення мікроконтролера [30].

Arduino може використовуватися для розробки інтерактивних систем, керованих різними датчиками і перемикачами. Такі системи, в свою чергу, можуть управляти роботою різних індикаторів, двигунів та інших пристроїв. Проекти Arduino можуть бути як самостійними, так і взаємодіяти з програмним забезпеченням, що працює на персональному комп'ютері (наприклад, додатками Flash, Processing, MaxMSP). Будь-яку плату Arduino можна зібрати вручну або ж купити готовий пристрій; Середовище розробки для програмування такої плати має відкритий вихідний код і повністю безкоштовна.

Існує безліч інших мікроконтролерів і мікропроцесорних пристроїв, призначених для програмування різних апаратних засобів: Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard і багато інших. Всі ці пристрої пропонують схожу функціональність і покликані звільнити користувача від необхідності заглиблюватися в дрібні деталі внутрішнього устрою мікроконтролерів, надавши йому простий і зручний інтерфейс для їх програмування. Arduino також спрощує процес роботи з мікроконтролером, але на відміну від інших систем надає ряд переваг для викладачів, студентів і радіоаматорів:

- Низька вартість. У порівнянні з схожими апаратними платформами, плати Arduino мають відносно низьку вартість: готові модулі Arduino коштують не дорожче 50 \$, а можливість зібрати плату вручну дозволяє максимально заощадити кошти і отримати Arduino за мінімальну ціну;

- Кросплатформеність. Програмне забезпечення Arduino працює на операційних системах Windows, Macintosh OSX і Linux, в той час, як більшість подібних систем орієнтовані на роботу тільки в Windows.

- Просте та зручне середовище програмування. Середовище програмування Arduino зрозуміле і просте для початківців, але при цьому досить гнучке для досвідчених користувачів. Воно основане на середовищі програмування Processing, що може бути зручно для викладачів. Завдяки цьому, студенти, які вивчають програмування в середовищі Processing, зможуть легко освоїти Arduino.

- Розширюване програмне забезпечення з відкритим вихідним кодом. Програмне забезпечення Arduino має відкритий вихідний код, завдяки цьому досвідчені програмісти можуть змінювати і доповнювати його. Можливості мови Arduino можна також розширювати за допомогою C++ бібліотек. Завдяки тому, що він заснований на мові AVR C, просунуті користувачі, охочі розібратися в технічних деталях, можуть легко перейти з мови Arduino на C або вставляти ділянки AVR- C коду безпосередньо в програми Arduino.

- Розширюване відкрите апаратне забезпечення. Пристрої Arduino побудовані на базі мікроконтролерів Atmel ATmega8 і ATmega168. Завдяки тому, що всі схеми модулів Arduino опубліковані під ліцензією Creative Commons, досвідчені інженери і розробники можуть створювати свої версії пристроїв на основі існуючих. І навіть звичайні користувачі можуть збирати дослідні зразки Arduino для кращого розуміння принципів їх роботи і економії коштів [31].

Arduino Nano – це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328 (Arduino Nano 3.0) або ATmega168 (Arduino Nano 2.x), адаптований для використання з макетної плати. За функціональністю пристрій схожий на Arduino Duemilanove, і відрізняється від нього розмірами, відсутністю роз'єму живлення, а також іншим типом (Mini-B) USB-кабелю. Arduino Nano розроблено і випускається фірмою Gravitech [32].

Arduino Nano може живитися через кабель Mini-B USB, від зовнішнього джерела живлення з нестабілізованою напругою $6\div 20\text{В}$ (через вивід 30) або зі стабілізованою напругою 5 В (через вивід 27). Пристрій автоматично вибирає джерело живлення з більшою напругою.

Напруга на мікросхему FTDI FT232RL подається тільки в разі живлення Arduino Nano через USB. Тому при живленні пристрою від інших зовнішніх джерел (НЕ USB), вихід 3,3 В (формований мікросхемою FTDI) буде неактивний, в результаті чого світлодіоди RX і TX можуть мерехтіти при наявності високого рівня сигналу на виводах 0 і 1.

Обсяг пам'яті програм мікроконтролера ATmega168 становить 16 кБ (з них 2 кБ використовуються завантажувачем); в ATmega328 – цей обсяг становить 32 кБ (з яких 2 кБ також відведені під завантажувач). Крім цього, ATmega168 має 1 кБ оперативної пам'яті SRAM і 512 байт EEPROM (для взаємодії з якої служить бібліотека EEPROM); а мікроконтролер ATmega328 – 2 кБ SRAM і 1 кБ EEPROM.

З використанням функцій `pinMode ()`, `digitalWrite ()` і `digitalRead ()` кожен з 14 цифрових виводів Arduino Nano може працювати в якості входу або виходу. Робоча напруга виводів – 5 В. Максимальний струм, який може віддавати або споживати один вивід, становить 40 мА. Всі виводи пов'язані з внутрішніми резисторами (за умовчанням відключеними) номіналом $20\div 50\text{ кОм}$.

Крім основних, деякі виводи Arduino можуть виконувати додаткові функції:

- Послідовний інтерфейс: виводи 0 (RX) і 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці виводи з'єднані з відповідними виводами мікросхеми-перетворювача USB-UART від FTDI.

- Зовнішні переривання: виводи 2 і 3. Ці виводи можуть бути налаштовані в якості джерел переривань, що виникають при різних умовах: при низькому рівні сигналу, по фронту, по спаду або при зміні сигналу.

- ШІМ: виводи 3, 5, 6, 9, 10 і 11. За допомогою функції `analogWrite ()` можуть виводити 8-бітові аналогові значення в вигляді ШІМ-сигналу.

- Інтерфейс SPI: виводи 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Дані виводи дозволяють здійснювати зв'язок по інтерфейсу SPI. У пристрої реалізована апаратна підтримка SPI, проте на даний момент мова Arduino поки її не підтримує.

- Світлодіод: вивід 13. Вбудований світлодіод, приєднаний до цифрового виводу 13. При відправці значення HIGH світлодіод включається, при відправці LOW — вимикається. В Arduino Ethernet є 8 аналогових входів, кожен з яких може уявити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За умовчанням, вимір напруги здійснюється щодо діапазону від 0 до 5 В. Проте, верхню межу цього діапазону можна змінити, використовуючи вивід AREF і функцію `analogReference()`. Крім цього, деякі з виводів мають додаткові функції:

- I2C: вивід 4 (SDA) і 5 (SCL). З використанням бібліотеки `Wire` (документація на веб-сайті `Wiring`) дані виводи можуть здійснювати зв'язок по інтерфейсу I2C (TWI). Крім перерахованих на платі існує ще кілька виводів:

- AREF. Опорна напруга для аналогових входів. Може бути задіяний функцією `analogReference()`.

- Reset. Формування низького рівня (LOW) на цьому висновку призведе до перезавантаження мікроконтролера. Зазвичай цей вивід служить для функціонування кнопки скидання на платах розширення[33].

Arduino Nano надає ряд можливостей для здійснення зв'язку з комп'ютером, ще одним Arduino або іншими мікроконтролерами. У ATmega168 і ATmega328 є приймач UART, що дозволяє здійснювати зв'язок з послідовним інтерфейсів за допомогою цифрових виводів 0 (RX) і 1 (TX). Мікросхема FTDI FT232RL забезпечує зв'язок приймача з USB-портом комп'ютера, і при підключенні до ПК дозволяє Arduino визначатися як віртуальний COM-порт (драйвера FTDI включені в пакет програмного забезпечення Arduino). У пакет програмного забезпечення Arduino також входить спеціальна програма, що дозволяє зчитувати і відправляти на Arduino прості текстові дані. При передачі даних комп'ютера через USB на платі будуть мигати світлодіоди RX і TX. (При послідовній передачі даних за допомогою виводів 0 і 1 дані світлодіоди задіюються) [34].

Бібліотека `SoftwareSerial` дозволяє реалізувати послідовну зв'язок на будь-яких цифрових виводах Arduino Nano.

У мікроконтролерах ATmega328 і ATmega168 також реалізована підтримка послідовних інтерфейсів I2C (TWI) і SPI. У програмне забезпечення Arduino входить бібліотека `Wire`, що дозволяє спростити роботу з шиною I2C [35].

3.2 Особливості середовища розробки Arduino IDE

Платформа Arduino являє собою сімейство мікроконтролерів на базі процесорів Atmel, STM і ARM. Всі мікроконтролери програмуються на мові C/C++ в середовищі розробки Arduino IDE. Велика частина плат Arduino (крім малих, таких як micro, pro mini) мають ідентичне розташування пінів і дозволяють підключати уніфіковані сторонні модулі, названі Шілд (Shield). На всіх платах є набір цифрових і аналогових пінів, а так само інтерфейси SPI і I2C. Для роботи зі сторонніми модулями в середовищі розробки є менеджер бібліотек, куди зібрані найбільш часто використовувані для Arduino бібліотеки [38].

Arduino – апаратна обчислювальна платформа для конструювання, основними компонентами якої є плата мікроконтролера з елементами введення/виведення. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: Processing, Adobe Flash, Max/MSP, Pure Data, SuperCollider). Інформація про плату (рисунок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі.

IDE (інтегроване середовище розробки) – це спеціальна програма, що працює на вашому комп'ютері і дозволяє вам писати скетчі для плати Arduino. Коли ви натиснете кнопку вивантаження скетчу на плату, то код, який ви написали, буде транслюватися в мову C/C++, потім буде переданий компілятору avr-gcc, який і проведе фінальну трансляцію в код, зрозумілий мікроконтролеру [39].

Цикл програмування Arduino спрощено виглядає так:

- Підключення плати до USB-порту комп'ютера;
- Напишіть програму-скетч;
- Вивантажуйте цей скетч на плату через USB-з'єднання і зачекайте кілька секунд для перезапуску плати;
- Плата виконає написаний вами скетч.

На рис. 3.1 зображена панель інструментів середовища розробки Arduino IDE.

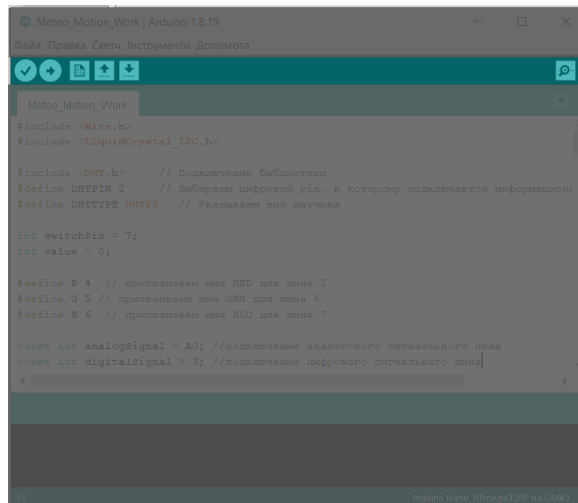


Рисунок 3.1 –Панель інструментів середовища розробки Arduino IDE

На панелі інструментів відображаються такі значки: Перевірити, Вивантажити, Створити, Відкрити та Зберегти.

-Перевірити: компілює та запускає код. Далі завантажує код на підключену плату. Перш ніж завантажити скетч, ми повинні переконатися, що вибрано правильну плату та порт;

- Вивантажити: знадобиться USB-з'єднання для з'єднання плати і комп'ютера. Останні плати Arduino можна скинути автоматично перед початком вивантаження. У старих платах потрібно було натиснути кнопку Reset, яка є на ній. Як тільки вивантаження завершиться успішно, можемо помітити миготіння світлодіодів Tx і Rx. Якщо вивантаження не вдається, у вікні помилки відобразиться повідомлення;

-Створити: використовується для створення нового ескізу або відкриття нового вікна;

- Відкрити: використовується для відкриття вже створеного файлу. Вибраний файл буде відкрито в поточному вікні;

-Зберегти: використовується для збереження поточного скетчу або коду;

-Монітор послідовного порту: розташований в правому куті панелі інструментів. Коли ми відкриваємо монітор послідовного порту, плата перезавантажиться.

На рис. 3.2 зображене повідомлення програми в середовищі розробки Arduino IDE.

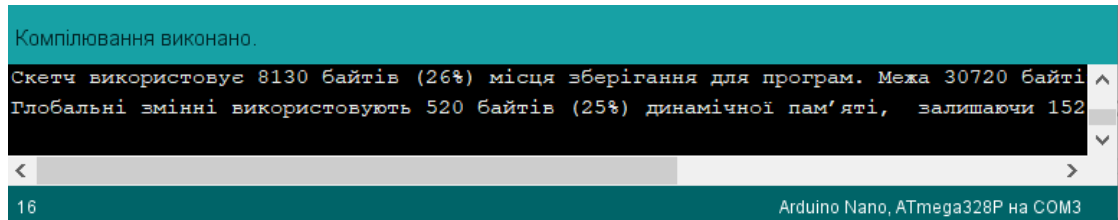


Рисунок 3.2 – Повідомлення програми середовища розробки Arduino IDE

Повідомлення програми – програма показує вам помилки кодів і деякі проблеми, які можуть виникнути під час процесу програмування. І пояснює вам тип помилки чи проблеми, яка сталася та її причину. І в ньому представлені деякі інструкції, які ви повинні застосувати, щоб усунути помилку чи проблему.

3.3 Програмна та апаратна реалізація системи охоронної сигналізації на платформі Arduino

Плата Arduino Nano V3.0 AVR ATmega328P – аналог флагманської Uno у мініатюрному розмірі. На ній передбачено все необхідне для зручної роботи з мікроконтролером: 14 цифрових входів/виходів, 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм Mini-B USB, роз'єм живлення, роз'єм для внутрішньосхемного програмування (ICSP) та кнопка скидання.

Arduino Nano може бути заживлена від Mini-B USB роз'єму або зовнішнього джерела живлення 6÷12В (пін "Vin") або 5 В стабільного зовнішнього живлення (пін "5V"). Живлення автоматично перемикається на джерело з вищою напругою.

Плата Arduino Nano V3.0 AVR ATmega328P має такі характеристики:

- мікроконтролер ATmega328P;
- тип корпусу TQFP-32;
- робоча напруга 5 В;
- вхідна напруга (рекомендована) 7÷12 В;
- цифрових входів/виходів 14 (з яких 6 можуть бути використані як ШІМ);
- аналогових входів 8;
- сила струму на входах/виходах 40 мА;
- сила струму для 3,3 В виходу 50 мА;

- пам'ять 32 кБ, з яких 2кБ використовується бутлоадером;
- SRAM 2 кБ;
- EEPROM 1 кБ;
- частота 16 МГц.

На рис. 3.3 подано схему розташування виводів плати Arduino Nano V3.0 AVR ATmega328P.

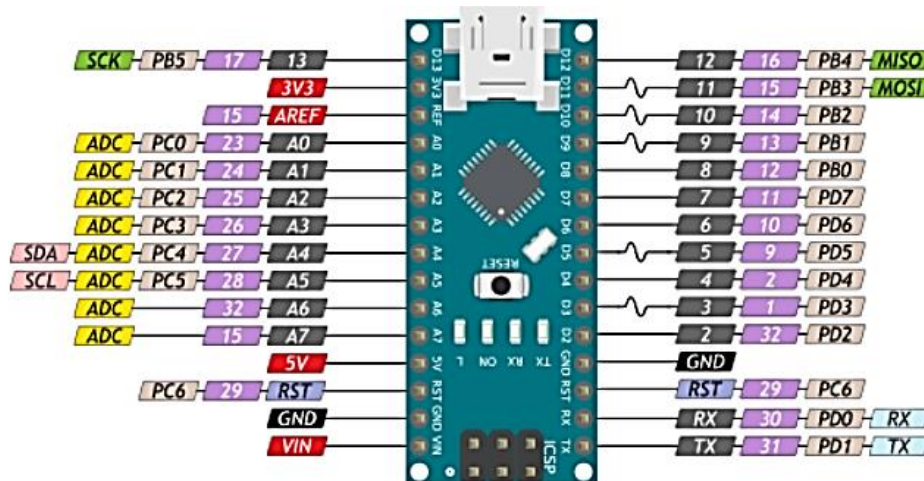


Рисунок 3.3 – Схема розташування виводів плати Arduino Nano V3.0 AVR ATmega328P

- VIN вхідний пін для підключення зовнішнього джерела живлення з напругою від 7 до 12 вольт.
- 5V вихідний пін від регулятора напруги на платі з виходом 5 вольт та максимальним струмом 800 мА. Живити пристрій через пін 5V не рекомендується, оскільки це може пошкодити плату.
- 3.3V вихідний пін від стабілізатора мікросхеми FT232R з виходом 3,3 вольт та максимальним струмом 50 мА. Живити пристрій через пін 3.3V не рекомендується, оскільки це може пошкодити плату.
- GND піни заземлення (нульовий потенціал).
- AREF пін для підключення зовнішньої опорної напруги АЦП (аналогово-цифрового перетворювача), відносно якої здійснюються аналогові вимірювання під час використання функції `analogReference()` з параметром `EXTERNAL`.

Порти введення/виведення. Цифрові входи/виходи:

- Піни 0-13 використовуються як цифрові входи та виходи.
- Логічний рівень "1" — 5 В.

- Логічний рівень "0" — 0 В.
- Максимальний струм виходу — 40 мА.

До контактів підключені підтягувальні резистори, які за замовчуванням вимкнені, але можуть бути активовані програмно.

- Піни 3, 5, 6, 9, 10 та 11 дозволяють виводити аналогові значення у вигляді ШІМ-сигналу (широтно-імпульсна модуляція). Розрядність ШІМ становить 8 біт і не може бути змінена.

- Піни А0-А7 дозволяють перетворювати аналогову напругу у цифрове значення. Розрядність АЦП (аналогово-цифровий перетворювач) становить 10 біт і не змінюється. Діапазон вхідної напруги — від 0 до 5 В. Подання напруги понад 5 В може пошкодити мікроконтролер.

- Піни А4 (SDA) та А5 (SCL) використовуються для зв'язку з периферійними пристроями за інтерфейсом I²C. Для роботи використовується бібліотека Wire.

- Піни 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) використовуються для зв'язку з периферійними пристроями за інтерфейсом SPI (Послідовний периферійний інтерфейс). Для роботи використовується бібліотека SPI.

- Піни 0 (RX) та 1 (TX) використовуються для прийому (RX) та передачі (TX) даних через послідовний інтерфейс UART (Універсальний асинхронний приймач/передавач).

Цифровий датчик температури та вологості підвищеної точності. Датчик DHT22 має заводське калібрування і характеризується низьким енергоспоживанням.

Датчик вологості та температури DHT22 має такі характеристики [40]:

- діапазон вимірювання вологості 0-100%;
- точність 0.1 °С;
- діапазон вимірювання температури -40 ~ 80 °С;
- точність вимірювання вологості ± 2% RH;
- точність вимірювання температури ± 0.5 градуса;
- напруга живлення 3.6-6 В;
- кількість виводів 4;
- ультранизьке енергоспоживання.

На рис. 3.4 подано схему підключення датчика вологості та температури DHT22.

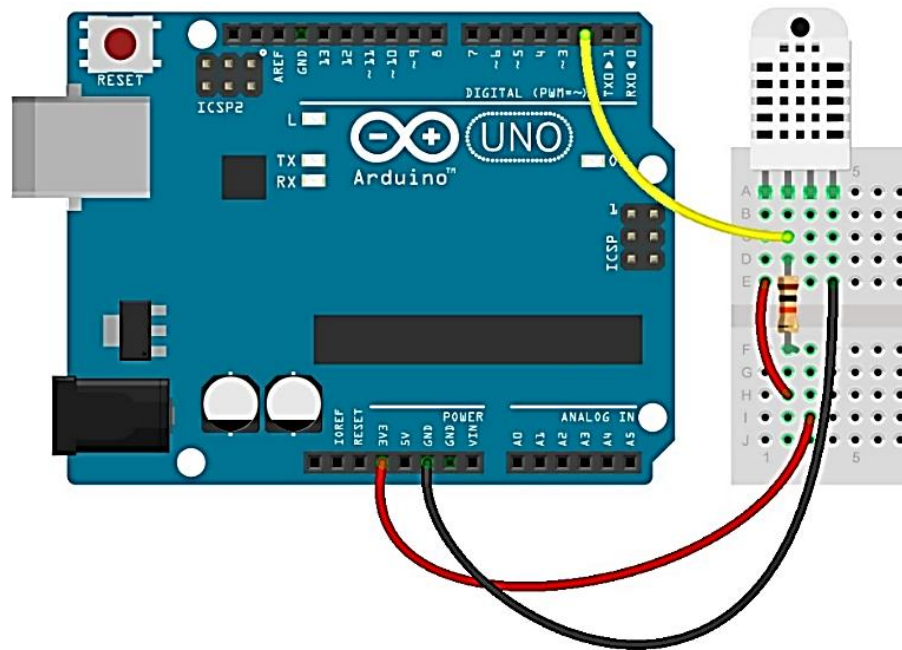


Рисунок 3.4 – Схема підключення датчика вологості та температури DHT22

Для вимірювання вологості створюється змінна:

```
float h = dht.readHumidity();// змінна для зчитування значення
вологості
```

Для вимірювання температури створюються змінна:

```
float t = dht.readTemperature();// змінна для зчитування значення
температури
```

Далі перевіряється підключення датчика та вивід помилки в послідовний порт в разі проблем з підключенням.

```
if (isnan(h) || isnan(t)) {
    lcd.print(F("Failed to read from DHT sensor!"));
    return;
}
```

Модуль датчика якості повітря MQ135 призначений для визначення вмісту та кількості шкідливих і небезпечних газів в повітрі таких як: NH₃, NO_x, пари алкоголю, бензину, диму, CO₂ тощо. Модуль побудований на датчику MQ135.

Модуль датчика якості повітря MQ-135 має такі характеристики [41]:

- аналоговий вихід датчика 0÷5 В;

- напруга живлення модуля 5 В постійного струму;
- розміри 32 мм X22 мм X27 мм.

На рис. 3.5 подано схему підключення датчика якості повітря MQ135.

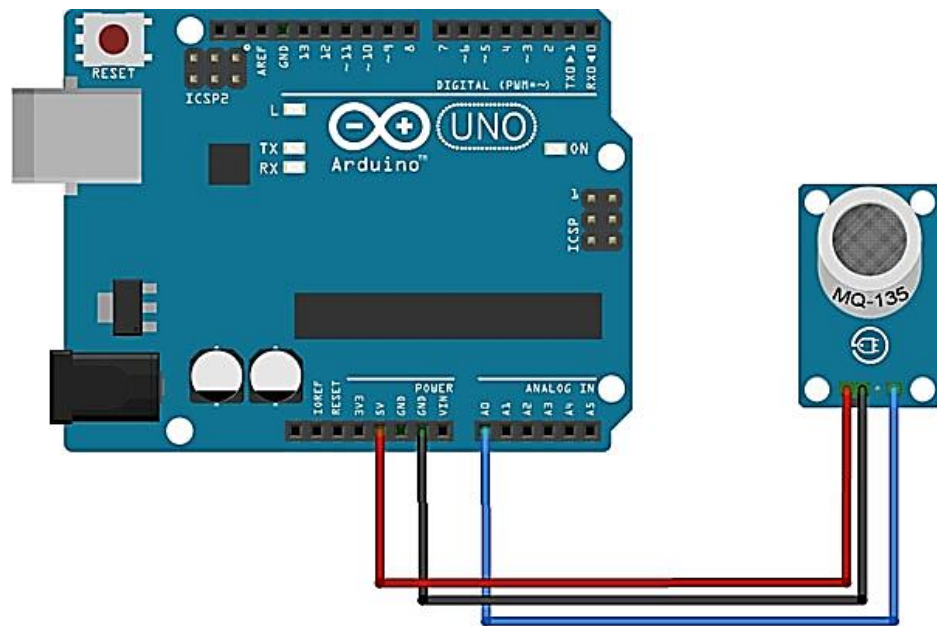


Рисунок 3.5 – Схема підключення датчика якості повітря MQ135

Для роботи з датчиком MQ-135 спочатку зчитуємо присутність газу, який вимірюємо, в даному випадку CO₂ (Вуглекислий газ) та його кількість. Виведення кількості газу реалізується таким чином:

```
noGas = digitalRead(digitalSignal); //зчитуємо
значення про присутність газу
gasValue = analogRead(analogSignal); // зчитуємо
значення кількості газу
```

Модуль RGB-світлодіода KY-016 призначений для використання в моделях і пристроях, які використовують платформу Arduino. На одній матриці розміщується відразу три кристали: червоний, зелений і синій. Завдяки незалежному керуванню такими кристалами, можна отримати будь-який колір. Щоб ці кристали не виділялися точково через лінзу RGB – світлодіода, на корпус може наноситися матове покриття, яке згладжує колір. Коли прикладається напруга до RGB світлодіода, відповідні напівпровідникові матеріали починають випромінювати світло. Діоди R, G та B генерують світло різної довжини хвилі, що відповідає їхнім основним кольорам червоний, зелений та синій. Для отримання бажаного кольору

світла RGB світлодіода використовується принцип змішування цих трьох основних кольорів. Якщо подати напругу тільки на червоний діод, то буде виділятися червоне світло. Те саме стосується зеленого і синього діодів. Але, комбінування різних співвідношень інтенсивності червоного, зеленого та синього світла дозволяє отримати весь спектр кольорів.

Модуль RGB світлодіода має такі характеристики [42]:

- номінальний струм 20 мА;
- максимальний імпульсний струм 50 мА;
- пряма номінальна напруга (червоний) 2 В;
- пряма номінальна напруга (зелений) 3,4 В;
- пряма номінальна напруга (синій) 3,4 В;
- максимальна зворотна напругу 5 В.

На рис. 3.6 подано схему підключення RGB-світлодіода.

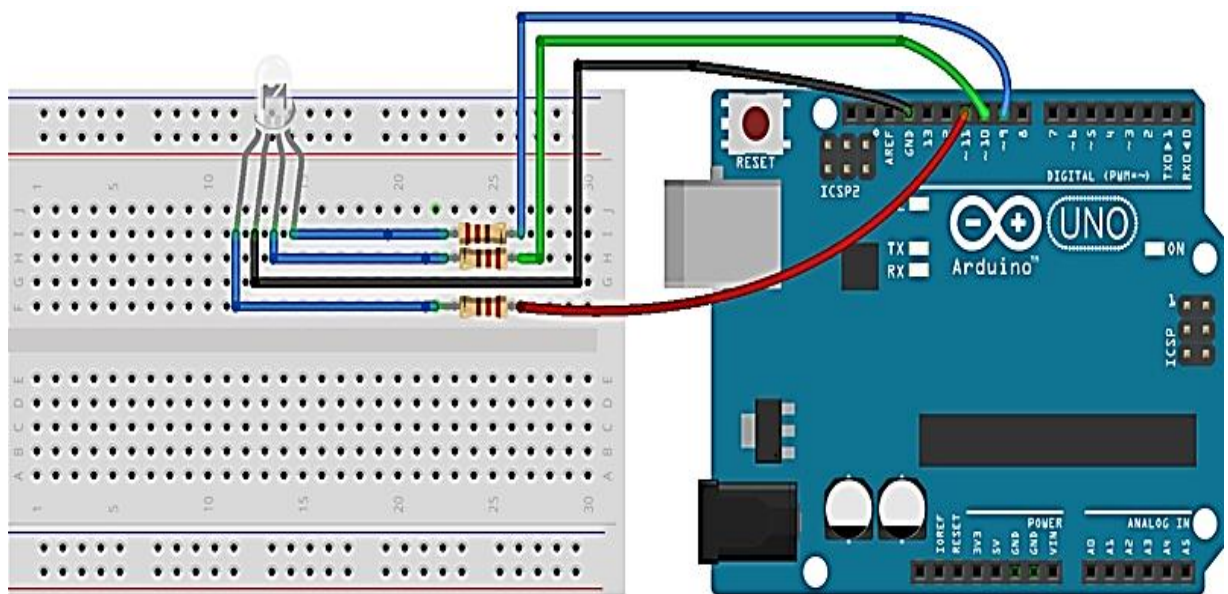


Рисунок 3.6 – Схема підключення RGB-світлодіода

Наприклад, якщо система знята з під охорони – світлодіод світиться блакитним, якщо система під охороною - – світлодіод світиться зеленим, якщо тривога - червоним. Реалізується це таким чином:

```
void alarmMode() {
    digitalWrite(buzzPin, LOW); // on buzzer
```

```

digitalWrite(Red, HIGH);
digitalWrite(Green, LOW);
digitalWrite(Blue, LOW);
...
}

```

Для затримки на екрані використовується функція `delay()`; яка призупиняє виконання програми на задану в параметрі кількість мілісекунд.

Інфрачервоний датчик руху HC-SR501 для Arduino та інших мікроконтролерів дозволяє виявляти рух людини або свійської тварини на відстані до 7 метрів (можна регулювати). Має два входи живлення (+5 В і Земля) та один цифровий вихід, яким можна знімати дані. Якщо перешкод немає – на ньому буде високий рівень (3,3 В), якщо є низький (0 В).

Якщо перемикача встановлена в положення Н, то на виході буде високий рівень весь час, поки датчик буде вловлювати рух, якщо стан L, то стан виходу буде перемикатися з високого на низький і навпаки приблизно 1 раз в секунду.

Інфрачервоний датчик руху HC-SR501 має такі характеристики [43]:

- дальність виявлення 0 - 7 м;
- кут спрацьовування 110 ° на дистанції до 7 м;
- напруга живлення (рекомендована) 4.5÷12 В;
- вихідна напруга логічного рівня 0 ÷ 3.3 В;
- час затримки 0.3÷300 секунд (регулюється);
- метод спрацьовування L неповторне перемикання; Н повторюване перемикання;
- максимальний вихідний струм 65 мА;
- робочі температури -20 – +50 град, Цельсія;
- розміри 32×24 мм.

На рис. 3.7 подано схему підключення ІЧ датчика руху HC-SR501.

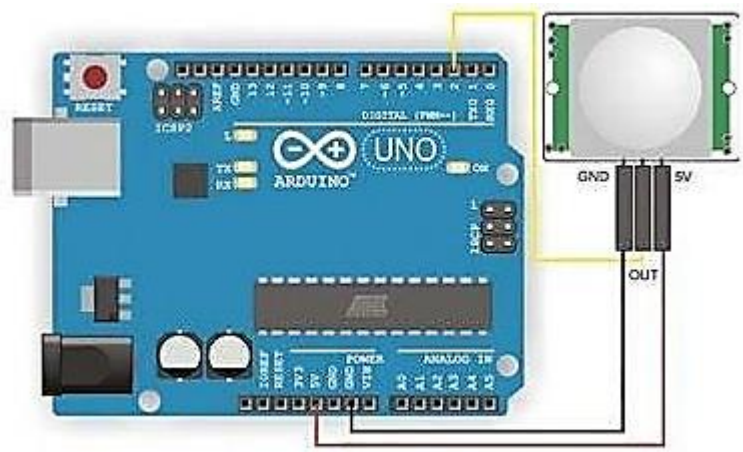


Рисунок 3.7 – Схема підключення ІЧ датчика руху HC-SR501

Для роботи ІЧ датчиком руху HC-SR501 присвоюємо для switchPin значення 8.

```
int switchPin = 8;
```

Далі встановлюємо режим роботи, а саме INPUT:

```
pinMode(switchPin, INPUT);
```

Використовуємо функцію digitalRead, яка дає змогу зчитати дані з піна:

```
value = digitalRead(switchPin);
```

Модуль магнітного датчика Холла А3144. Датчик Холла це цифровий однополярний датчик, що працює на ефекті Холла, суть якого полягає в тому, що при приміщенні в магнітне поле деякого провідника з постійним струмом, в цьому провіднику виникає поперечна різниця потенціалів.

Технічні характеристики магнітного датчика Холла А3144 [44]:

- робоча напруга 4.5-24 В;
- вихід цифровий;
- тип магнітний;
- індикація LED.

На рис. 3.8 подано схема підключення модуля магнітного датчика Холла А3144.

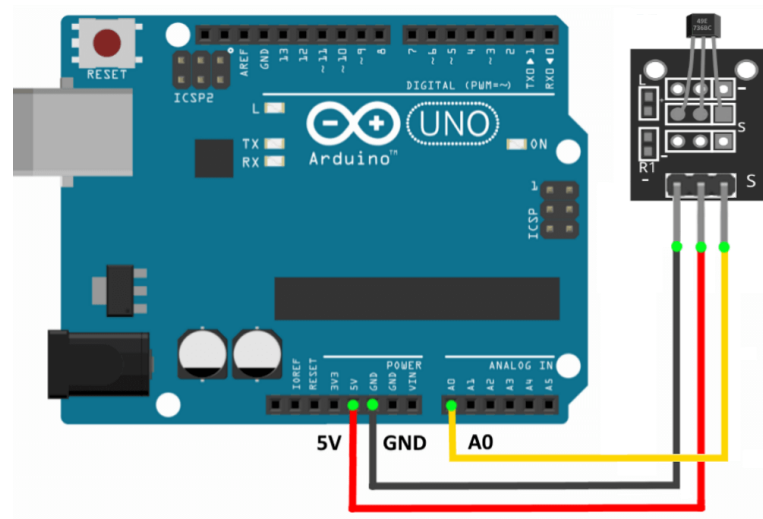


Рисунок 3.8 – Схема підключення модуля магнітного датчика Холла А3144

Модуль з динаміком (Low-level). Живлення 3.3 - 5V. Три виводи: живлення, земля і сигнал. Розміри плати 32 x 13 мм. Є кріпильний отвір.

Активний зумер - найпростіший модуль для отримання звуку частотою близько 2 кГц.

Для його роботи достатньо подати на нього напругу 5V, так як усередині модуля є власний генератор. Споживаний струм - 50 мА[45]. На рис. 3.9 подано схему підключення динаміка.

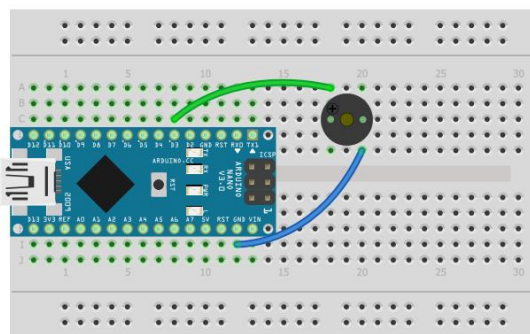


Рисунок 3.9 – Схема підключення динаміка

NodeMCU являє собою плату розробника на базі чіпа ESP8266 (версія ESP12E), який являти собою UART-WiFi модуль з ультра низьким споживанням. Сам чіп проєктувався для пристроїв зі світу інтернет речей, а дана плата дозволяє спростити розробку, тому що на ній вже реалізовано підключення по USB, регулятор живлення і всі виводи чіпа розведені на гребінки зі стандартним кроком 2.54 мм, що дозволяє вставити його в макетну плату і

створити прототип навіть не включаючи паяльник. Крім цього плата поставляється з прошивкою NodeMCU.

Характеристики Wi-Fi модуля NodeMCU V3 ESP8266 [46]:

- WiFi 802.11 b / g / n;
- підтримка STA / AP / STA + AP режимів;
- вбудований стек протоколів TCP / IP з підтримкою множинних клієнтських підключень (до 5);
- D0 ~ D8, SD1 ~ SD3 можуть бути використані як GPIO, PWM, ІІС;
- ток на виведення 15 мА;
- AD0 1 виведення АЦП;
- живлення 4.5 - 9В (10В максимум), живлення від USB з наданням отладного інтерфейсу;
- споживання обмін даними: ~ 70 мА (200 мА максимум), очікування: <200 мкА;
- швидкість передачі 110-460800 б/сек;
- підтримка UART / GPIO інтерфейсів передачі даних;
- перепрошивка з хмари або через USB;
- відстань між контактними пинами 28 мм;
- діапазон робочих температур -40 ~ +125 ° С;
- вага 18 г.

На рис. 3.10 подано зовнішній вигляд Wi-Fi модуля NodeMCU V3 ESP8266.



Рисунок 3.10 – Зовнішній вигляд Wi-Fi модуля NodeMCU V3 ESP8266

5В-ий 1-канальний модуль реле низького рівня (low level), вимагає 5-20мА для спрацьовування, тобто може управлятися безпосередньо з виводів мікроконтролера Arduino або подібних. Вмикається логічним нулем, вимикається логічною одиницею. На модулі є два світлодіода: червоний сигналізує про наявність напруги живлення, зелений - про спрацьовування реле.

Характеристики релейного модуля Arduino 1 канального реле 5В [46]:

- Максимальний струм комутації реле 10А при 250В
- На платі є світлодіод що сигналізує про поточний статус реле
- Розміри 44x17x17мм

На рис. 3.11 подано зовнішній вигляд релейного модуля Arduino 1 канального реле 5В.



Рисунок 3.11 – Зовнішній вигляд релейного модуля Arduino 1 канального реле 5В

3.4 Тестування роботи системи охоронної сигналізації та аналіз результатів

На рис. 3.12 наведено зовнішній вигляд системи охоронної сигналізації, наочно демонструє апаратну реалізацію розробленої системи охоронної сигналізації. На зображенні видно основні компоненти системи.

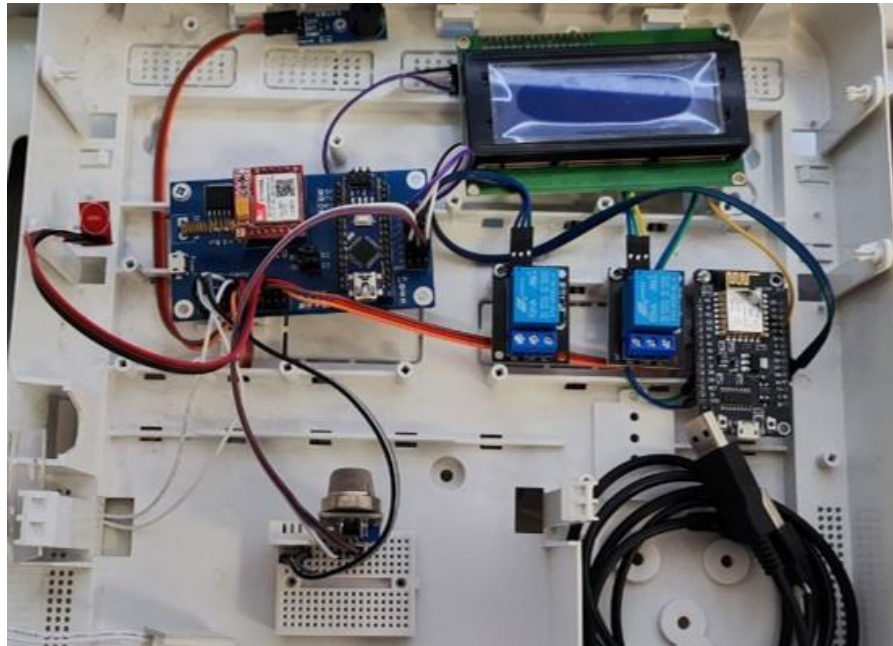


Рисунок 3.12 – Зовнішній вигляд системи охоронної сигналізації

Arduino Nano може житись від Mini-B USB роз'єму або зовнішнього джерела живлення 6-12В (пін "Vin") або 5В стабільного зовнішнього живлення (пін "5V"). Живлення автоматично перемикається на джерело з більш високою напругою. Контролер з WiFi може житись від Micro-USB роз'єму або зовнішнього джерела живлення 4.5-9 В (пін "Vin") або 3.3 В стабільного зовнішнього живлення (пін "3.3V"). Я використовую подачу живлення через Mini-B USB роз'єм для головного контролера та живлення через Micro-USB роз'єм для контролер з WiFi від павербанку або ПК. На рис 3.13 показано, охоронну сигналізацію яка знаходиться в стані знято з-під охорони. Після підключення живлення до системи, головний контролер ініціалізує всі підключені датчики, модулі зв'язку та виконавчі пристрої. Система починає збирати дані з датчиків, потім мікроконтролер аналізує зібрані дані виводить на дисплей та додаток, при старті система знаходиться в стані знято з-під охорони до дій користувача. В стані знято з-під охорони система не повідомляє і не формує тривогу виявлення руху або відкриття дверей, але контролює стан дверей для подальшої заборони постановки під охорону, при порушенні дверей.

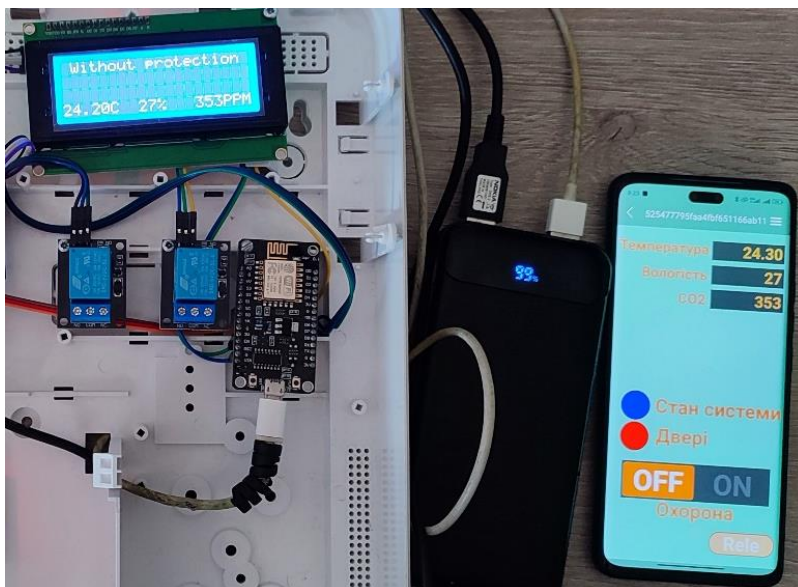


Рисунок 3.13 – Система охоронної сигналізації в стані знята з-під охорони

В системі охоронної сигналізації передбачено захист від постановки під охорону під час відкритих дверей та вивід на дисплей попередження про неможливість поставити сигналізацію під охорону, для переведення сигналізації в стан під охороною потрібно щоб двері були зачинені. Це інформаційне повідомлення виведене при постановці під охорону сенсорною кнопкою.

На рис 3.14 показано вивід помилки про відчинені двері та неможливість поставити під охорону систему.



Рисунок 3.14 – Вивід помилки про неможливість поставити під охорону через відкриті двері

В системі охоронної сигналізації передбачено різні способи керування охороною. Можливо поставити під охорону локально, за допомоги сенсорної кнопки, також керувати дистанційно, за допомоги додатка або через SMS. Коли система охоронної сигналізації переводиться в стан під охороною, вона активує всі свої захисні функції, готові до виявлення будь-яких несанкціонованих дій. На дисплей виводиться інформаційне повідомлення про стан системи, на додатку змінюється індикатор стану системи, виводиться стан дверей, та перемикач переходить у стан увімкнено. Система розпочинає зчитувати сигнали від датчика руху, та датчика відкриття дверей, при порушенні – переходить в стан тривоги.

На рис 3.15 показано, охоронну сигналізацію яка знаходиться в стані поставлено під охорону.

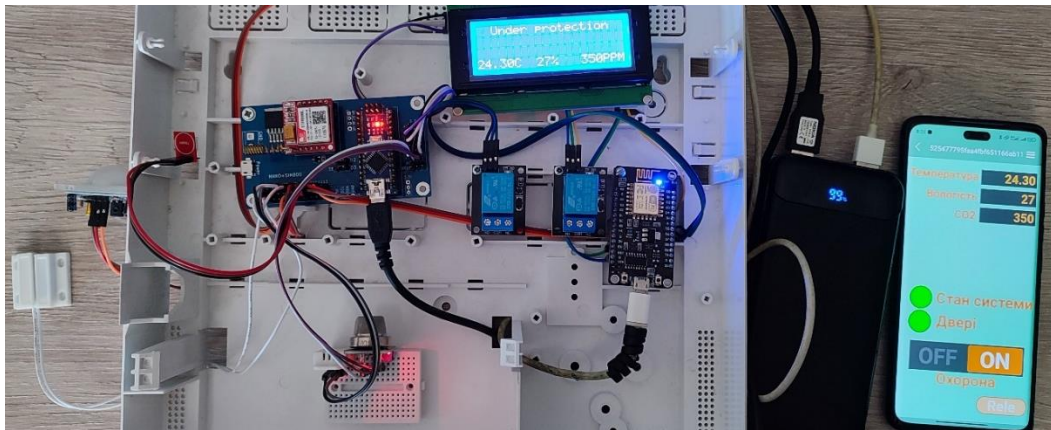


Рисунок 3.15 – Система охоронної сигналізації в стані поставлено під охорону

Перехід від стану поставлено під охорону можливий в стан знято з-під охорони користувачем, за допомогою додатка, SMS чи локально кнопкою, в стан тривоги можливий перехід після спрацювання датчика руху та/або спрацювання датчика відкриття дверей. На додатку стан системи переходить у червоний індикатор, в залежності від стану дверей змінюється індикатор стану дверей, на телефон приходить SMS, що система в тривозі, вмикається звуковий сповіщувач, та виводиться на дисплей стан тривоги і які датчики були порушені. Датчик руху має дальність виявлення до 7 метрів та кут огляду 110 градусів. Було проведено тести з різними кутами підходу та відстанями до датчика від 5 сантиметрів, до 7 метрів. Результати підтвердили стабільне виявлення руху в межах заявлених характеристик. Система забезпечує своєчасне спрацювання на рух, час реакції

датчиків складає менш ніж 1 секунду. Тестування проводилося шляхом порушення датчика руху та датчика відкриття дверей, у всіх випадках система відразу змінювала свій стан на стан тривоги та відпрацьовувала сповіщення користувача.

На рис 3.16 показано, охоронну сигналізацію яка знаходиться в стані тривоги.

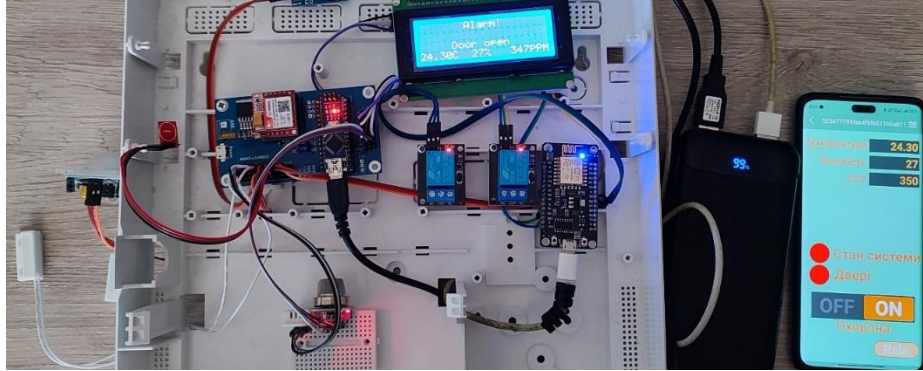


Рисунок 3.16 – Система охоронної сигналізації в стані тривоги

На рис 3.17 показано, знімок екрану спілкування системи по GSM-каналі, за допомогою SMS користувач має змогу змінити стан систему, чи побачити зміну стану яка відбулася локально чи дистанційно за допомогою додатка. Якщо надіслати команду info на відповідь буде надіслана SMS з такою інформацією, як температура, відносна вологість, рівень CO₂ та стан системи.



Рисунок 3.17 – Знімок екрану спілкування системи по GSM-каналі

Додаток зроблений за допомогою сервіса RemoteXY – це система розробки та використання мобільних графічних інтерфейсів для керування контролерами зі смартфона або планшета. Конфігурація графічного інтерфейсу зберігається у контролері. Конфігурація графічного інтерфейсу завантажується на мобільний додаток з контролера. На рис 3.18, а) показано знімок першого екрану додатка для керування системою. У верхній частині знаходяться текстові поля для виведення інформації про мікроклімат у приміщенні. В середині екрану є 2 світлових індикатора які сповіщають про стан системи і стан датчика відкриття дверей, нижче розміщено перемикач стану системи, в ньому є 2 положення, під охороною та знято з-під охорони, в низу екрану є кнопка переходу на другий екрану додатка для керування реле. На другому екрану додатка є 2 перемикачі якими можна керувати незалежно один від одного і незалежно від стану системи, також в нижній частині є кнопка переходу на перший екран додатка.

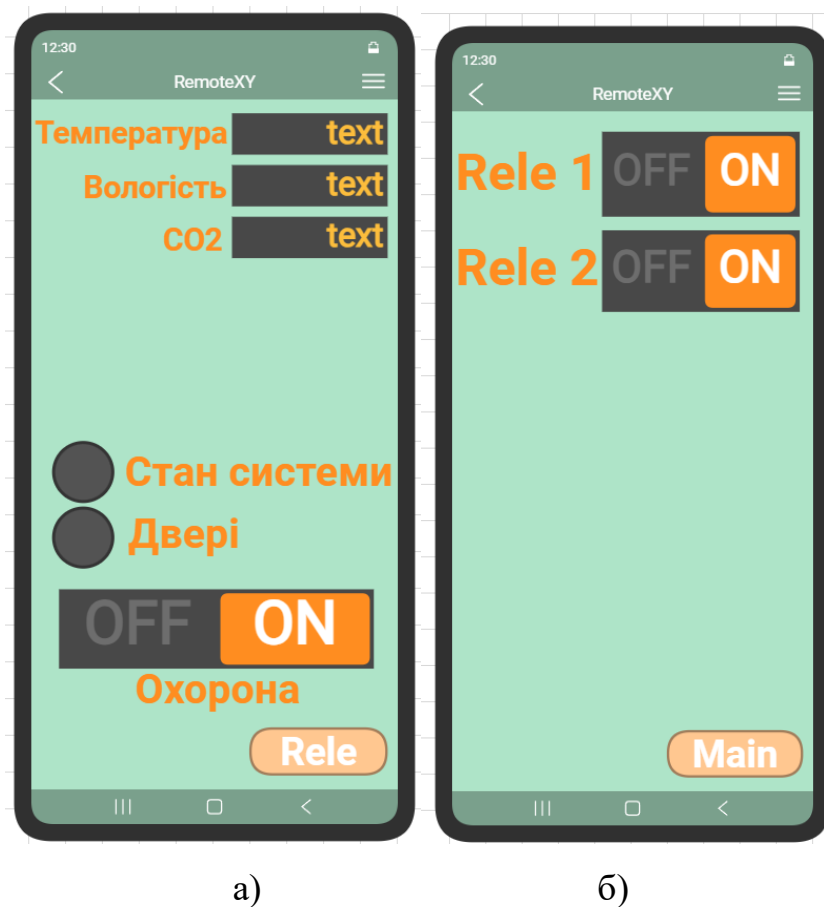


Рисунок 3.18 – Знімок екрану додатка: а - першого екрану для керування системою, б - другого екрану додатка для керування реле

Тестування мікрокліматичних датчиків, таких як: датчики температури, вологості та якості повітря виконувалось у приміщеннях із різними рівнями вологості та температури. Система успішно визначила зміну кліматичних параметрів та оновила інформацію на дисплеї в реальному часі, мікроконтролер точно фіксує зміни параметрів і передають дані користувачу.

Також тестувалися загроза витік газу та задимленість, спрацювання датчика якості повітря. Визначення витік газу відбувається при перевищенні порогу 500 PPM(частка на мільйон), тоді як дим фіксується при перевищенні 700 PPM, пожежа у свою чергу фіксується при перевищенні 1000PPM та різкому збільшенню температури більше 40°C. У разі виявлення витоку газу, диму, пожежі система виводить відповідне повідомлення на дисплеї із вказівкою на рівень PPM та тип загрози. Було проведено тести для перевірки роботи системи при різних рівнях PPM, вологості та температурі. У тестах використовувалися контрольовані джерела газу, такі як запальничка та контрольовані джерела диму - тліючий папір. Тестування системи відбувалося за умовами нормальної температури в межах від 20 °C до 30°C, та відносній вологості від 20% до 50%.

На рис 3.19 показано вивід на дисплей повідомлення про витік газу. Видно, що значення якості повітря більше порогового, тобто більше 500PPM, показники температури та вологості знаходяться в нормі.

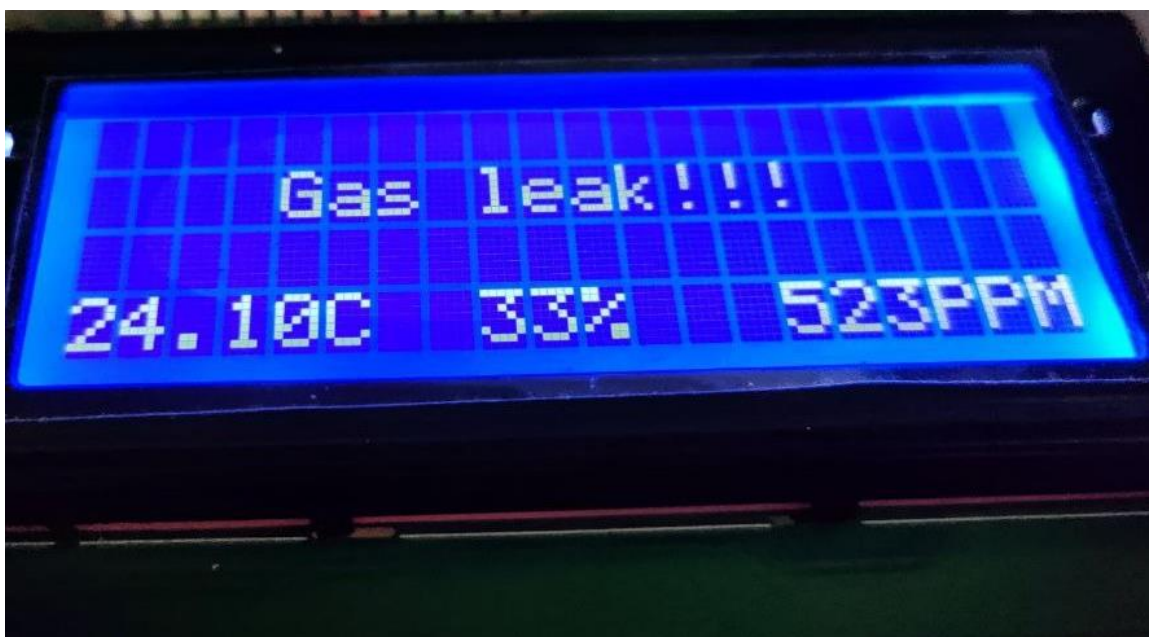


Рисунок 3.19 – Вивід на дисплей повідомлення про витік газу

На рис 3.20 показано вивід на дисплей повідомлення про виявлення диму. Видно, що значення якості повітря більше порогового, тобто більше 700PPM, показники температури та вологості знаходяться в допустимих значеннях.



Рисунок 3.20 – Вивід на дисплей повідомлення про витік газу

3.5 Висновок

В цьому розділі проведено аналіз вибору платформ для розробки програмно-апаратної системи. Проведено аналіз найчастіше використовуваних для розробки систем охоронних сигналізацій мов програмування: C++ та C. Обґрунтовано вибір мови програмування: а саме C++. Проаналізовано особливості середовища розробки Arduino IDE. Реалізовано програмну та апаратну частину системи охоронної сигналізації. Проведено тестування системи охоронної сигналізації на платформі Arduino. Система успішно виконує поставлені завдання. Датчики руху та відкриття дверей своєчасно реагують на несанкціонований доступ. Система оперативно повідомляє користувача про будь-які зміни стану за допомогою звукових сигналів, світлової індикації, SMS-повідомлень та мобільного додатку. Можливість керування системою за допомогою мобільного додатку забезпечує зручність та гнучкість у використанні. Датчики температури, вологості та якості повітря точно фіксують зміни параметрів і передають дані на головний контролер. Система оперативно реагує на зміни стану датчиків та формує відповідні сигнали.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного аудиту науково-технічної розробки

Метою проведення комерційного та технологічного аудиту є оцінювання потенціалу розробки системи охоронної сигналізації на основі платформи Arduino. У результаті оцінювання можна буде зробити висновок щодо особливостей організації подальшого впровадження розробки з врахуванням встановленого рейтингу.

Для проведення аудиту потрібно залучити трьох незалежних експертів. У випадку цієї конкретної розробки це керівник магістерської кваліфікаційної роботи Куперштейн Л. М., начальник відділу розробки охоронних систем ТОВ «Тірас-12» Дячук О.О. та спеціаліст у галузі охоронних систем ТОВ«Тірас-12» Кошкар'єв Б. П.

Оцінку комерційного потенціалу розробки системи охоронної сигналізації на основі платформи Arduino буде здійснено за дванадцятьма критеріями, згідно рекомендацій. Результати оцінювання комерційного потенціалу заноситься до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного успіху системи охоронної сигналізації на основі платформи Arduino.

Критерії	Експерти		
	Куперштейн Л. М.	Дячук О.О.	Кошкар'єв Б. П.
	Бали, виставлені експертами		
1	5	5	5
2	2	3	3
3	5	5	5
4	1	2	3
5	4	5	5

Продовження таблиці 4.1

6	5	5	5
7	1	1	1
8	4	5	5
9	4	4	4
10	5	5	5
11	5	5	5
12	3	3	3
Сума балів	44	48	49
Середньоарифметична сума балів, СБ	47		

За даними таблиці 4.1 можна зробити висновок щодо рівня комерційного потенціалу розробки системи охоронної сигналізації на основі платформи Arduino. При цьому потрібно користуватись рекомендаціями, наведеними в таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41 - 48	Високий
31 – 40	Вищий середнього
21 – 30	Середній
11 – 20	Нижчий середнього
0 - 10	Низький

Таким чином, можна зробити висновок, що рівень комерційного потенціалу розробки відповідає високому рівню. Це досягається за рахунок покращення захищеності приміщення від загроз несанкціонованого проникнення та техногенних загроз за рахунок розробки та впровадження охоронної сигналізації.

Конкурентоспроможність розкривається через систему якісних та економічних показників.

Якісні показники конкурентоспроможності характеризують властивості розробки, завдяки яким вона задовольняє конкретні потреби.

Економічні показники конкурентоспроможності характеризують сумарні витрати споживачів на задоволення їх потреб цією розробкою. Вони складаються з витрат на придбання і витрат, пов'язаних з експлуатацією виробу. Оцінювання рівня конкурентоспроможності науково-технічної розробки здійснюється в декілька етапів

Процедура визначення якісних одиничних параметричних індексів за технічними показниками здійснюється за відповідними формулами. Якщо збільшення величини параметра свідчить про підвищення якості нової розробки, одиничний параметричний індекс розраховується за формулою:

$$q_i = \frac{P_i}{P_{\text{базі}}}.$$

Якщо зменшення величини параметра свідчить про підвищення якості нової розробки, то одиничний параметричний індекс розраховується за оберненою формулою:

$$q_i = \frac{P_{\text{базі}}}{P_i},$$

де q_i – одиничний параметричний індекс, розрахований за i -м параметром;

P_i – значення i -го параметра розробки;

$P_{\text{базі}}$ – аналогічний параметр базової розробки-аналога, з якою проводиться порівняння.

$$\text{Кількість датчиків} - q_1 = \frac{38}{60} = 0.63;$$

$$\text{Функціонал} - q_2 = \frac{44}{80} = 0.55;$$

$$\text{Точність виміру температури} - q_3 = \frac{90}{25} = 3.6;$$

$$\text{Ціна} - q_4 = \frac{92}{31} = 2.9.$$

Таблиця 4.3 - Технічні та економічні параметри аналога та нової науково-технічної розробки

Параметр	Одиниця виміру	Аналог	Нова розробка	Індекс зміни значення параметру	Коефіцієнт вагомості
Функціонал	Шт.	60	38	0.63	0.7
Кількість датчиків	Шт.	44	80	0.55	0.5
Точність виміру температури	%	90	25	3.6	1
Ціна	Грн.	92	31	2.9	1

Груповий показник конкурентоспроможності за нормативними параметрами розраховується як добуток частинних показників за кожним параметром за формулою:

$$I_{\text{гп}} = \prod_{i=1}^n q_i ,$$

де $I_{\text{гп}}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i -м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

Якщо хоч один з частинних показників дорівнює 0, то розробка неконкурентоспроможна.

$$I_{\text{гп}} = 0.63 \cdot 0.55 \cdot 3.6 \cdot 2.9 = 3.61$$

Значення групового параметричного індексу за технічними параметрами визначається з урахуванням вагомості кожного параметра:

$$I_{\text{гп}} = \sum_{i=1}^n q_i \cdot \alpha_i ,$$

де $I_{\text{гп}}$ – груповий параметричний індекс за технічними показниками;

q_i – одиничний параметричний показник i -го параметра;

α_i – вагомість i -го параметричного показника, $\sum_{i=1}^n \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

$$I_{\text{ТП}} = (0.63 \cdot 0.7) + (0.55 \cdot 0.5) + (3.6 \cdot 1) + (2.9 \cdot 1) = 7,2$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розраховують інтегральний показник конкурентоспроможності за формулою:

$$K_{\text{ІІТ}} = I_{\text{ІІП}} \cdot \frac{I_{\text{ТП}}}{I_{\text{ЕП}}}.$$

На основі інтегрального показника формується висновок про конкурентоспроможність оцінюваної розробки. У разі $K_{\text{ІІТ}} < 1$ аналізована розробка поступається базовому зразку за конкурентоспроможністю, за $K_{\text{ІІТ}} > 1$ – перевищує зразок. За умови рівної конкурентоспроможності $K_{\text{ІІТ}} = 1$. Однак потрібно мати на увазі, що за зростання ІТП показник $K_{\text{ІІТ}}$ збільшується, характеризуючи зростання конкурентоспроможності. За зростання ІЕП показник $K_{\text{ІІТ}}$ зменшується, відображаючи зниження конкурентоспроможності.

$$K_{\text{ІІТ}} = 3.61 \cdot \frac{7.2}{2.9} = 8.96$$

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

4.2.1 Витрати на оплату праці

Основна заробітна плата кожного із розробників, якщо вони працюють в установах бюджетної сфери:

$$Z_o = \frac{M}{T_p} \cdot t,$$

де M – місячний посадовий оклад розробника, грн.

У 2024 році мінімальний розмір окладу 8000 грн., враховуючи надбавки рекомендовано брати величину у розмірі 10000-14000 грн.;

T_p – число робочих днів в місяці;

t – число робочих днів роботи розробника. Зроблені розрахунки заносяться до таблиці 4.3.

Таблиця 4.3 – Заробітна плата розробників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	21000	954	5	4 770
Інженер-програмного забезпечення	24000	1 090	25	27 250
Інженер з тестування	12000	545	12	6 540
Всього:				38 560

Основна заробітна плата робітників Z_p , яких вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Z_p = \sum_{i=1}^n t_i \cdot C_i,$$

де t_i - норма часу (трудомісткість) на виконання конкретної роботи, годин; n - число робіт по видах та розрядах; C_i - погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. C_i визначається за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}},$$

де M_M – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати, грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 - 23$ дні;

$t_{зм}$ – тривалість зміни, год.

Таблиця 4.4 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год.	Розряд роботи	Тарифний коефіцієнт	Погодинна тариф. ставка, грн	Величина, грн
Розробка	20	7	2.2	110	2200
Налаштування	1	5	1.7	85	85
Монтаж	2	3	1.35	67,5	135
Всього					2420

Додаткова заробітна плата $Z_{\text{дод}}$ всіх розробників, що брали участь у виконанні роботи, розраховується як 10-12% від суми основної заробітної плати всіх робітників, та розраховується за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%},$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Тобто розрахунок буде виглядати як:

$$Z_{\text{дод}} = (38560 + 2420) \frac{10}{100} = 4098.$$

4.2.2 Відрахування на соціальні заходи

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення. Нарухування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_{\text{н}} = (З_{\text{о}} + З_{\text{р}} + З_{\text{доод}}) \cdot \frac{Н_{\text{зп}}}{100\%},$$

де $H_{\text{зп}}$ – норма нарахування на заробітну плату.

$$З_{\text{н}} = (38560 + 2420 + 4098) \cdot \frac{22}{100} = 9917$$

4.2.3 Амортизація програмних засобів

До статті «Амортизація програмних засобів» відносять амортизаційні відрахування по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, за його наявності в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12},$$

де $Ц_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{в}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років. Проведені розрахунки необхідно звести до таблиці 4.5.

Таблиця 4.5 - Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
ПК	44000	3	3	3 666
Всього				3 666

4.2.4 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень.

Витрати на матеріали (М), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{vj},$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{vj} – вартість відходів j -го найменування, грн/кг.

Таблиця 4.6 - Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за одиницю, грн	Норма витрат, шт	Вартість витраченого матеріалу, грн
Папір канцелярський офісний (A4)	230, 00	1,0	230
Папір для заміток (A5)	115, 00	4,0	460
Начиння канцелярське	175, 00	1,0	175
Всього			865

Таблиця 4.7 - Витрати на компоненти системи

Найменування компонента	Кількість	Вартість, грн	Загальна вартість
Arduino Nano V3.0 AVR ATmega328P TYPE-C	1	179	179
GSM модуль на SIM800L	1	200	200
ІЧ датчик руху для Arduino HC-SR501	1	46	46
Модуль RGB світлодіода KY-016 для Arduino	1	12	12
Модуль датчика якості повітря MQ-135	1	70	70
Датчик вологості та температури DHT22	1	210	210
LCD 2004 I2C символний дисплей 20x4 (синій)	1	277	277
Модуль з динаміком активний (buzzer)	1	24	24
TTP223 TOUCH KEY сенсорний датчик для Arduino	1	10	10
Модуль датчика Холла А3144 цифровий	1	29	29
Модуль реле 5В 10А з опторозв'язкою (low level)	2	39	78
Wi-Fi модуль NodeMCU V3 ESP8266	1	149	149
Всього			1284

4.2.5 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i},$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$B_e = 12,5 \cdot 0,6 \cdot 100 \cdot 0,8 = 600$ грн.

4.2.6 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_v = (3_o + 3_p) \cdot \frac{H_{iv}}{100\%},$$

де H_{iv} – норма нарахування за статтею «Інші витрати».

$$I_v = (38560 + 2420) \cdot \frac{55}{100} = 22\,539 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = 3_o + 3_p + 3_{\text{доп}} + 3_n + M + K_s + B_{\text{стел}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сп}} + I_v + B_{\text{нзс}}.$$

До статті «Накладні витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

$$\begin{aligned} B_{\text{заг}} &= 38560 + 2420 + 4098 + 9917 + 3\,666 + 865 + 1\,412,4 + 600 + 22\,539 \\ &= 84\,077 \end{aligned}$$

4.2.7 Загальні витрати

Загальні витрати ЗВ на завершення науково-дослідної роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta},$$

де η – коефіцієнт, який характеризує етап виконання науково-дослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta = 0,1$; технічного проектування, то $\eta = 0,2$; розробки конструкторської документації, то $\eta = 0,3$; розробки технологій, то $\eta = 0,4$; розробки дослідного зразка, то $\eta = 0,5$; розробки промислового зразка, то $\eta = 0,7$; впровадження, то $\eta = 0,9$.

$$ЗВ = \frac{84\,077}{0.4} = 210\,192,5 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Система охоронної сигналізації на основі платформи Arduino» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів які придбали систему у періоди часу, що аналізуються;

N - кількість споживачів які користуються системою у році до впровадження результатів нової науково-технічної розробки, прийmemo 20000 осіб;

$Ц_6$ - вартість системи, прийmemo 1700 грн;

$\pm\Delta\Pi_o$ - зміна вартості послуги від впровадження результатів, приймемо зростання на + 200,00 грн.

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	100	250	155	95

Можливе збільшення чистого прибутку у потенційного інвестора для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,82$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. Рекомендується брати $\rho = 0,2 \dots 0,5$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$.

Π_o – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, $\Pi_o = \Pi_b \pm \Delta\Pi_o$;

$$\Delta\Pi_1 = (200 \cdot 20000 + 1900 \cdot 100) \cdot 0,82 \cdot 0,4 \cdot \left(1 - \frac{18}{100}\right) = 1126942,4$$

грн.

$$\Delta\Pi_2 = (200 \cdot 20000 + 1900 \cdot 250) \cdot 0,82 \cdot 0,4 \cdot \left(1 - \frac{18}{100}\right) = 1203596$$

грн.

$$\Delta\Pi_3 = (200 \cdot 20000 + 1900 \cdot 155) \cdot 0,82 \cdot 0,4 \cdot \left(1 - \frac{18}{100}\right) =$$

1155048,72 грн.

$$\Delta\Pi_4 = (200 * 20000 + 1900 * 95) * 0,82 * 0.4 * \left(1 - \frac{18}{100}\right) = 1124387,28 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t},$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках) від моменту початку впровадження науковотехнічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{1126942,4}{(1+0,1)^1} + \frac{1203596}{(1+0,1)^2} + \frac{1155048,72}{(1+0,1)^3} + \frac{1124387,28}{(1+0,1)^4} = 1024493,091 + 1203596 + 1270553,592 + 1360508,609 = 4859151,292 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B,$$

де $K_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $\text{інв } k = 2 \dots 5$, але може бути і більшим;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 210\,192,5 = 420\,385 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV,$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн.

Якщо величина $E_{абс}$ буде мати велике додатне значення, то це може свідчити про потенційну зацікавленість інвесторів у впровадженні та комерціалізації цієї науково-технічної розробки. Але для остаточного прийняття рішення про впровадження науково-технічної розробки та виведення її на ринок цього недостатньо.

$$E_{абс} = 4859151,292 - 420\,385 = 4\,438\,766,292 \text{ грн.}$$

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно. Внутрішня економічна дохідність інвестицій E_v , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науковотехнічної розробки, розраховується за формулою:

$$E_v = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1,$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, роки.

$$E_v = \sqrt[4]{1 + \frac{4\,438\,766,292}{420\,385}} - 1 = 0.84$$

Далі визначають бар'єрну ставку дисконтування мін τ , тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій мін τ визначається за формулою:

$$\tau_{\min} = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,1 \dots 0,12$;

f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05 \dots 0,5$, але може бути і значно вищою.

Якщо величина $E_v > \tau_{\min}$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

$$\tau_{\min} = 0,10 + 0,4 = 0,5$$

Далі розраховуємо період окупності інвестицій Ток (DPP, Discounted Payback Period), які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_v},$$

де E_v – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = \frac{1}{0,84} = 1,19.$$

Якщо $T_{ок} < 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.4 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система охоронної сигналізації на основі платформи Arduino» становить 47 балів, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 8.96 рази.

Також термін окупності становить 1.19 р., що менше 3-х років, що свідчить про комерційну вдалість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Система охоронної сигналізації на основі платформи Arduino». Така робота є достатньо перспективною розробкою, та має шанси на комерціалізацію.

ВИСНОВКИ

Відповідно до поставлених завдань дослідження були досягнуті наступні результати: обґрунтовано доцільність розробки системи охоронної сигналізації на основі платформи Arduino, проведений аналіз сучасних комерційних систем охоронних сигналізацій дозволив виявити їхні основні недоліки. Використання платформи Arduino дозволяє створити економічно ефективну, гнучку та адаптивну систему, яка відповідає індивідуальним потребам користувачів. Було досліджено існуючі системи охоронних сигналізацій та їхні характеристики. Аналіз існуючих рішень дозволив визначити оптимальну конфігурацію системи на основі платформи Arduino, а також обрати необхідні компоненти та датчики. Було виділено ключові функції, які повинна виконувати система, зокрема: виявлення несанкціонованих дій, контроль параметрів мікроклімату, передача сповіщень та можливість віддаленого керування. У процесі проектування була створена детальна структурна та монтажна схема системи, яка включає всі необхідні компоненти для забезпечення функціональності також були визначені основні елементи системи. Здійснено моделювання логіки роботи системи за допомогою діаграм станів та діаграм комунікації. Визначено основні сценарії взаємодії між компонентами та алгоритми обробки подій. Програмно реалізовано систему охоронної сигналізації на основі платформи Arduino. Реалізовано функції зчитування даних з датчиків, обробки отриманої інформації, формування сповіщень та взаємодії з мобільним додатком через Wi-Fi або взаємодії з користувачем через GSM-зв'язок. Виконано тестування програми та проведено аналіз отриманих результатів проведення тестування системи охоронної сигналізації підтвердило її здатність виконувати поставлені завдання. Датчики руху та відкриття дверей своєчасно реагували на несанкціонований доступ, система оперативно інформувала користувача за допомогою звукових сигналів, світлової індикації, SMS-повідомлень та мобільний додаток. Система забезпечила стабільну роботу у режимах моніторингу, оповіщення та дистанційного керування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. V Міжнародній науково-практичній конференції «Інформаційна безпека та інформаційні технології» <https://indico.ldubgd.edu.ua/event/55/> (дата звернення 10.12.2024)
2. Квартирні крадіжки в Україні у 2023 році <https://opendatabot.ua/analytics/home-burglaries> (дата звернення 10.12.2024)
3. Охоронна сигналізація <https://ukrinfosystems.com.ua/uk/design-and-construction/alarm-system> (дата звернення 10.12.2024)
4. Поліція Охорони про крадіжки з будинків та надійний захист <https://okna.ua/ua/library/politsiya-okhorony-pro-kradizhky-z> (дата звернення 10.12.2024)
5. TIRAS Technologies <https://tiras.technology/> (дата звернення 10.12.2024)
6. Jablotron <https://www.jablotron.com/en/> (дата звернення 10.12.2024)
7. Paradox <https://www.paradox.com/> (дата звернення 10.12.2024)
8. DSC <https://www.dsc.com/> (дата звернення 10.12.2024)
9. Honeywell <https://www.honeywellhome.com/us/en/products/security/> (дата звернення 10.12.2024)
10. Bosch <https://www.boschsecurity.com/xs/en/> (дата звернення 10.12.2024)
11. Satel <https://www.satel.eu/ru/cat/357> (дата звернення 10.12.2024)
12. U-Prox <https://u-prox.systems/> (дата звернення 10.12.2024)
13. Hikvision <https://hik.ua/> (дата звернення 10.12.2024)
14. Ajax <https://ajax.systems/ua/> (дата звернення 10.12.2024)
15. Arduino Nano <https://doc.arduino.ua/ru/hardware/Nano> (дата звернення 10.12.2024)
16. ESP8266 <https://drukarnia.com.ua/articles/esp8266-klyuch-do-bud-yakogo-routera-rXj2D> (дата звернення 10.12.2024)
17. GSM модуль на SIM800L <https://arduino.ua/prod1665-gsm-modyl-na-sim800l> (дата звернення 10.12.2024)
18. Датчик MQ135 <https://arduino.ua/ru/prod1201-modyl-datchika-kachestva-vozduha-mq135> (дата звернення 10.12.2024)

19. Датчик Холла <https://myproject.com.ua/a3144-modul-datchika-holla-ua.html>(дата звернення 10.12.2024)
20. ІЧ датчик руху для Arduino HC-SR501 <https://arduino.ua/prod193-ik-datchik-dvijeniya-dlya-arduino-hc-sr501>(дата звернення 10.12.2024)
21. ТТР223 <https://arduino.ua/prod264-ttp223-touch-key-sensornii-datchik-dlya-arduino>(дата звернення 10.12.2024)
22. Датчик вологості та температури DHT22 <https://arduino.ua/prod301-datchik-vlajnosti-i-temperatyri-dht22>(дата звернення 10.12.2024)
23. LCD 2004 I2C символний дисплей 20x4 <https://arduino.ua/prod663-lcd-2004-i2c-simvolnii-displei-20x4-sinii>(дата звернення 10.12.2024)
24. Модуль RGB світлодіода KY-016 <https://arduino.ua/prod2966-modyl-rgb-svetodioda-ky-016-dlya-arduino>(дата звернення 10.12.2024)
25. Модуль з динаміком активний (buzzer) <https://arduino.ua/prod490-modyl-s-dinamikom-buzzer>(дата звернення 10.12.2024)
26. Модуль реле 5В 10А високого рівня <https://arduino.ua/prod2888-modyl-rele-5v-10a-visokogo-urovnya-high-level>(дата звернення 10.12.2024)
27. Уніфікована мова моделювання (Unified Modeling Language - UML) <https://www.maxzosim.com/unifikovana-mova-modeluvannia/>(дата звернення 10.12.2024)
28. Використання уніфікованої мови моделювання (UML) http://iwanoff.inf.ua/oop_kn/LabTraining05.html(дата звернення 10.12.2024)
29. UML для бізнес-моделювання <https://evergreens.com.ua/ua/articles/uml-diagrams.html>(дата звернення 10.12.2024)
30. Arduino <https://vseplus.com.ua/article/cto-takoe-arduino-783>(дата звернення 10.12.2024)
31. Мікроконтролери і мікропроцесорні пристрої https://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.php(дата звернення 10.12.2024)
32. Arduino Nano <https://doc.arduino.ua/ru/hardware/Nano>(дата звернення 10.12.2024)

33. Arduino інтерфейси https://geekmatic.in.ua/ua/arduino_lesson_112(дата звернення 10.12.2024)
34. Arduino Nano https://uk.wikipedia.org/wiki/Arduino_Nano(дата звернення 10.12.2024)
35. Основи мікропроцесорної техніки https://ela.kpi.ua/bitstream/123456789/27992/1/OMPT_laboratorni.pdf(дата звернення 10.12.2024)
36. C++ http://www.znannya.org/?view=Cpp_basics(дата звернення 10.12.2024)
37. Основні оператори <http://tinker.uamper.com/docs/programuvannya-arduino.html>(дата звернення 10.12.2024)
38. Платформа Arduino <https://www.duet.edu.ua/uploads/DocS/9st9.pdf>(дата звернення 10.12.2024)
39. IDE (інтегроване середовище розробки) <https://apix-drive.com/ua/blog/useful/ide-i-sdk-strashnye-termyny-prostymi-slovami>(дата звернення 10.12.2024)
40. Датчик DHT22 <https://arduino.ua/ru/prod301-datchik-vlajnosti-i-temperatury-dht22>(дата звернення 10.12.2024)
41. Датчик MQ135 <https://arduino.ua/ru/prod1201-modyl-datchika-kachestva-vozdyha-mq135>(дата звернення 10.12.2024)
42. Модуль RGB-світлодіода <https://arduino.ua/ru/prod2966-modyl-rgb-svetodioda-ky-016-dlya-arduino> (дата звернення 10.12.2024)
43. Інфрачервоний датчик руху HC-SR501 <https://arduino.ua/ru/prod193-ik-datchik-dvijeniya-dlya-arduino-hc-sr501>(дата звернення 10.12.2024)
44. А3144 модуль датчик Хола [Електронний ресурс]. : URL: <https://myproject.com.ua/a3144-modul-datchika-holla-ua.html>(дата звернення 10.12.2024)
45. Модуль с динамиком активный (buzzer) [Електронний ресурс]. : URL: <https://arduino.ua/ru/prod490-modyl-s-dinamikom-buzzer>(дата звернення 10.12.2024)
46. Wi-Fi модуль NodeMCU V3 ESP8266 <https://arduino.ua/prod1492-wi-fi-modyl-nodemcu-esp8266>(дата звернення 10.12.2024)

47. Модуль реле 5В 10А високого рівня (high level)
<https://arduino.ua/prod2888-modyl-rele-5v-10a-visokogo-urovnya-high-level>(дата
звернення 10.12.2024)

ДОДАТКИ

ДОДАТОК А ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Система охоронної сигналізації на основі платформи Arduino
Автор роботи: Кутняк Максим Миколайович
Тип роботи: магістерська кваліфікаційна робота
Підрозділ кафедра захисту інформації ФІТКІ
Керівник Куперштейн Леонід Михайлович, к.т.н., доцент

Показники звіту подібності

Turnitin	
Оригінальність	74 %
Загальна схожість	26 %

Аналіз звіту подібності (відмітити потрібне):

- ☒ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

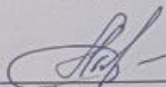
Заявляю, що ознайомлений з повним звітом подібності, який був згенерований Системою щодо роботи.

Автор роботи


(підпис)

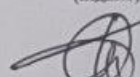
Кутняк М.М.
(прізвище, ініціали)

Особа, відповідальна за перевірку


(підпис)

Каплун В. А.
(прізвище, ініціали)

Керівник роботи


(підпис)

Куперштейн Л.М.
(прізвище, ініціали)

ДОДАТОК Б Код програми Arduino NANO

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <SoftwareSerial.h>

////////////////GSM////////////////
SoftwareSerial SIM800serial (2, 3);
#define OK 1
#define NOTOK 2
#define TIMEOUT 3

String txt;

unsigned long previousListMillis = 0;
const long intervalList = 2000; // Interval for "AT+CMGL=\"ALL\""
unsigned long currentMillis;
////////////////GSM////////////////

#define DHTPIN 6 // Выбираем цифровой pin, к которому подключается информационный выход датчика
#define mc38 7 // пін для МС-38 (датчика відкриття дверей)
#define motionPin 8 // присваиваем имя датчика руху пина 8
#define buzzPin 9 // пін для buzzer
#define sensorPin 10 // пін для сенсорної кнопки
#define Red 11 // присваиваем имя Red для пина 5
#define Green 12 // присваиваем имя Green для пина 6
#define Blue 13 // присваиваем имя Blue для пина 7

#define mq135Pin A1

#define DHTTYPE DHT22 // Указываем тип датчика
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,20,4);

byte systemState; // 0 - знято з-під охорони \ 1 - під охороною \ 2 - тривога
unsigned long lastButtonReadTime = 0; // змінна для збереження часу останнього зчитування кнопки
unsigned long lastMotionReadTime = 0; // змінна для збереження часу останнього зчитування Motion
unsigned long doorStateStartTime = 0; // змінна для збереження часу останнього зчитування МС38
unsigned long climateMillis = 0; // змінна для збереження часу останнього зчитування DHT22

const long interval = 6000; // Інтервал у мілісекундах

byte humidity;
float temperature;
int gasValue;

bool doorOpen;
byte motionState;
byte alarmLog = 0;

byte previousDoorState; // Попередній стан датчика mc38

// Методи
void sensorButton();
void workWithSecurity(bool local);
void doorState();
void sleepMode();
void protectionMode();
void alarmMode();
void displayedLCD(int location,int line, String text);
void climate();
void fireAlarm();
void parse_cmd( String str);
```

```

void sendSMS(String text);
byte SIM800command(String command, String response1, String response2, uint16_t timeOut, uint16_t repetitions);
byte SIM800waitFor(String response1, String response2, uint16_t timeOut);
String SIM800read();
// Методи

void setup() {
  Serial.begin(9600);
  //GSM//
  pinMode(5, OUTPUT);
  digitalWrite(5, HIGH); //SIM800 ON only for this module
  SIM800serial.begin(9600);
  delay(5000);
  SIM800command("AT", "OK", "ERROR", 500, 5);
  SIM800command("AT+CMGF=1", "OK", "ERROR", 2000, 1); //SMS to text mode - not PDU mode
  //GSM//
  lcd.init(); // initialize the lcd
  lcd.backlight();
  lcd.clear();

  dht.begin();

  pinMode(motionPin, INPUT);
  pinMode(mc38, INPUT);
  pinMode(sensorPin, INPUT);
  pinMode(buzzPin, OUTPUT);
  pinMode(Red, OUTPUT);
  pinMode(Green, OUTPUT);
  pinMode(Blue, OUTPUT);
  systemState = 0;
  digitalWrite(buzzPin, HIGH);
  Serial.println("Готовий до роботи!");
  displayedLCD(1,1,"Ready to work!");
}

void loop() {
  sensorButton(); // зчитування кнопки для зміни стану охорони
  doorState(); // зчитування стану датчика відкриття дверей
  climate(); // зчитування і вивід мікроклімату
  fireAlarm(); // зчитування та відправка смс якщо пожежа

  if(systemState == 0){
    sleepMode();
  }
  if(systemState == 1){
    protectionMode();
  }
  if(systemState == 2){
    alarmMode();
  }
  //////////////////////////////////GSM////////////////////////////////////
  currentMillis = millis();
  if (currentMillis - previousListMillis >= intervalList) {
    SIM800command("AT+CMGL=\"ALL\"", "OK", "ERROR", 5000, 1);///sms list
    parse_cmd(txt);
    SIM800command("AT+CMGD=1,3", "OK", "ERROR", 10000, 1);///sms delete
    previousListMillis = currentMillis;
  }
  //////////////////////////////////GSM////////////////////////////////////
}

void sensorButton(){
  unsigned long currentTime = millis();
  if (currentTime - lastButtonReadTime >= 1000) {
    if (digitalRead(sensorPin) == HIGH) {

```

```

    bool local = true;
    workWithSecurity(local);
}
lastButtonReadTime = currentTime;
}
}

void workWithSecurity(bool local){
    if (systemState == 0 && doorOpen == false){
        systemState = 1;
        if(local == true ){
            Serial.println("Постановка під охорону локально");
        }
        if(local == false ){
            Serial.println("Постановка під охорону дистанційно");
        }
        displayedLCD(1,0,"Under protection");
        displayedLCD(1,1,"");
        displayedLCD(1,2,"");
        sendSMS("Under protection");
    } else if (systemState == 1 || systemState == 2){
        systemState = 0;
        alarmLog = 0;
        if(local == true ){
            Serial.println("Знято з-під охорони локально");
        }
        if(local == false ){
            Serial.println("Знято з-під охорони дистанційно");
        }
        displayedLCD(1,0,"Without protection");
        displayedLCD(1,1,"");
        displayedLCD(1,2,"");
        sendSMS("Without protection");
    } else if (systemState == 0 && doorOpen == true) {
        systemState = 0;
        if(local == true ){
            Serial.println("Двері відкриті, неможливо поставити під охорону!");
        }
        if(local == false ){
            Serial.println("Двері відкриті, неможливо поставити під охорону!");
            sendSMS("Cannot be protected: Door open");
        }
        displayedLCD(1,1,"Cannot be protected");
        displayedLCD(1,2,"Door open");
    }
}

void doorState() {
    int doorState = digitalRead(mc38);
    // Перевірка, чи стан датчика змінився
    if (doorState != previousDoorState) {
        // Стан змінився, оновлюємо час початку нового стану
        doorStateStartTime = millis();
    }
    // Перевірка, чи датчик перебуває в одному стані протягом 3 секунд
    if (doorState == HIGH && (millis() - doorStateStartTime >= 2000)) {
        // Якщо датчик перебуває в стані HIGH протягом 3 секунд
        doorOpen = false;
        // Додайте код для обробки відкритих дверей тут
    } else {
        doorOpen = true;
    }
    previousDoorState = doorState;
}

```

```

//
void sleepMode(){
    digitalWrite(buzzPin, HIGH); // off buzzer
    digitalWrite(Red, LOW);
    digitalWrite(Green, LOW);
    digitalWrite(Blue, HIGH);
}

// режим охорони
void protectionMode() {

    digitalWrite(buzzPin, HIGH); // off buzzer
    digitalWrite(Red, LOW);
    digitalWrite(Green, HIGH);
    digitalWrite(Blue, LOW);

    unsigned long currentTime = millis();
    motionState = digitalRead(motionPin);
    if (currentTime - lastMotionReadTime >= 1000) {
        if (motionState == HIGH) {
            Serial.println("Motion");
            systemState = 2;
        }
        if (doorOpen == true){
            Serial.println("Door open");
            systemState = 2;
        }
        lastMotionReadTime = currentTime;
    }
}

// режим тривоги
void alarmMode() {
    digitalWrite(buzzPin, LOW); // on buzzer
    digitalWrite(Red, HIGH);
    digitalWrite(Green, LOW);
    digitalWrite(Blue, LOW);

    if (alarmLog < 1){
        displayedLCD(1,0,"Alarm!");
        Serial.println("Alarm!!!");
        sendSMS("Alarm!!!");
        alarmLog++;
    }
    unsigned long currentTime = millis();
    motionState = digitalRead(motionPin);
    if (currentTime - lastMotionReadTime >= 1000) {
        if (motionState == HIGH) {
            Serial.println("Motion");
            displayedLCD(1,1,"Motion");
        }
        if (doorOpen == true){
            Serial.println("Door open");
            displayedLCD(1,2,"Door open");
        }else {
            Serial.println("Door close");
            displayedLCD(1,2,"Door close");
        }
        lastMotionReadTime = currentTime;
    }
}

void displayedLCD(int location,int line, String text){
    lcd.setCursor(0, line);
    lcd.print("      ");
}

```

```

if (location == 0){
    lcd.setCursor(0,line);
}else if (location == 1){
    int lengthText = text.length();
    int spaces = (20 - lengthText) / 2;
    lcd.setCursor(spaces,line);
}else if (location == 2){
    int lengthText = text.length();
    int spaces = (20 - lengthText);
    lcd.setCursor(spaces,line);
}
lcd.print(text);
}

void climate(){
    unsigned long currentMillis = millis();
    humidity = dht.readHumidity();
    temperature = dht.readTemperature();
    gasValue = analogRead(mq135Pin);

    if (currentMillis - climateMillis >= interval){
        climateMillis = currentMillis;
        lcd.setCursor(0, 3);
        lcd.print(temperature);
        lcd.print("C");

        lcd.setCursor(8, 3);
        lcd.print(humidity);
        lcd.print("%");

        lcd.setCursor(14, 3);
        lcd.print(gasValue);
        lcd.print("PPM");
    }
}

void fireAlarm(){
    unsigned long currentMillis = millis();
    humidity = dht.readHumidity();
    temperature = dht.readTemperature();
    gasValue = analogRead(mq135Pin);

    if (currentMillis - climateMillis >= interval){
        climateMillis = currentMillis;
        if((gasValue > 1200) && (temperature > 40)){
            Serial.println("FIRE!!!!!!");
            String str = "Fire!!! Temperature: " + String(temperature) + "°C\r\nHumidity: " + String(humidity) + "%\r\nCO2: "
+ String(gasValue) + "PPM\r\nSystemState: " + String(systemState);
            sendSMS(str);
        }
    }
}

//////////////////////////////////GSM//////////////////////////////////
void parse_cmd( String str){
    String part1;
    Serial.println("read_cmd start");

    if (str.indexOf("ping")>-1) { //sms
        String str = "Hello! Arduino and SIM800 - work!"; //SMS text
        sendSMS(str);
    }
    if (str.indexOf("help")>-1) { //sms
        String strText = "help-list of commands\r\nping-Hello!\r\ninfo-system information(0-Without protection;1-Under
protection;2-Alarm)\r\nunder-Under protection\r\nwithout-Without protection"; //SMS text
    }
}

```

```

    sendSMS(strText);
}
if (str.indexOf("info")>-1) { //sms
    String str = "Temperature: " + String(temperature) + "*C\r\nHumidity: " + String(humidity) + "%\r\nCO2: " +
String(gasValue) + "PPM\r\nSystemState: " + String(systemState);
    sendSMS(str);
}
if (str.indexOf("under")>-1) { //sms
    systemState = 1;
    Serial.println("Under protection");
    displayedLCD(1,0,"Under protection");
    displayedLCD(1,1,"");
    displayedLCD(1,2,"");
    sendSMS("OK");
}
if (str.indexOf("without")>-1) { //sms
    systemState = 0;
    Serial.println("Without protection");
    displayedLCD(1,0,"Without protection");
    displayedLCD(1,1,"");
    displayedLCD(1,2,"");
    sendSMS("OK");
}
}

void sendSMS(String text){
    SIM800command("AT+CMGS=\""+380638835907\"",">", "ERROR", 5000, 1);
    String str = text;
    SIM800serial.print(str);//the content of the message
    char j=0x1A;
    SIM800command((String)j, ">", "ERROR", 2000, 1);
    Serial.println(text);
    delay(4000);
}

byte SIM800command(String command, String response1, String response2, uint16_t timeOut, uint16_t repetitions) {
    byte returnValue = NOTOK;
    byte countt = 0;

    while (countt < repetitions && returnValue != OK) {
        SIM800serial.println(command);

        if (SIM800waitFor(response1, response2, timeOut) == OK) {
            returnValue = OK;
        } else {returnValue = NOTOK;}
        countt++;
    }

    return returnValue;
}

byte SIM800waitFor(String response1, String response2, uint16_t timeOut) {
    uint16_t entry = 0;
    //uint16_t count = 0;
    String reply = SIM800read();
    byte retVal = 99;

    do {
        reply = SIM800read();
        delay(1);
        entry ++;
    } while ((reply.indexOf(response1) + reply.indexOf(response2) == -2) && entry < timeOut );

    if (entry >= timeOut) {
        retVal = TIMEOUT;
    }
}

```

[illegible]

ДОДАТОК В Код програми ESP8266

```
#include <SoftwareSerial.h>

////////////////////////////////////
// RemoteXY include library      //
////////////////////////////////////

// можете включить вывод отладочной информации в Serial на 115200
//#define REMOTEXY__DEBUGLOG

// определение режима соединения и подключение библиотеки RemoteXY
#define REMOTEXY_MODE__WIFI_CLOUD

#include <ESP8266WiFi.h>

// настройки соединения
#define REMOTEXY_WIFI_SSID "Pirania"
#define REMOTEXY_WIFI_PASSWORD "04022002n"
//#define REMOTEXY_WIFI_SSID "KTV"
//#define REMOTEXY_WIFI_PASSWORD "adtlab743345"
#define REMOTEXY_CLOUD_SERVER "cloud.remotexy.com"
#define REMOTEXY_CLOUD_PORT 6376
#define REMOTEXY_CLOUD_TOKEN "525477795faa4fbf651166ab112caaec"
#define REMOTEXY_ACCESS_PASSWORD "12345678"

#include <RemoteXY.h>

// конфигурация интерфейса RemoteXY
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] = // 302 bytes
{ 255,3,0,39,0,39,1,17,0,0,0,154,1,106,200,2,1,0,13,0,
  2,8,142,91,23,0,2,26,31,31,79,78,0,79,70,70,0,129,30,166,
  48,12,2,208,158,209,133,208,190,209,128,208,190,208,189,208,176,0,67,58,
  4,45,12,6,2,26,11,67,58,19,45,12,6,2,26,11,67,58,34,45,
  12,6,2,26,11,129,1,6,55,9,2,208,162,208,181,208,188,208,191,208,
  181,209,128,208,176,209,130,209,131,209,128,208,176,0,129,15,22,41,9,2,
  208,146,208,190,208,187,208,190,208,179,209,150,209,129,209,130,209,140,0,129,
  38,37,18,9,2,208,161,208,158,50,0,65,6,118,18,18,112,131,63,182,
  40,14,2,17,2,31,82,101,108,101,0,6,129,28,121,32,12,2,208,148,
  208,178,208,181,209,128,209,150,0,129,27,103,77,12,2,208,161,209,130,208,
  176,208,189,32,209,129,208,184,209,129,209,130,208,181,208,188,208,184,0,65,
  6,99,18,18,112,5,0,2,44,6,58,25,0,2,26,31,31,79,78,0,
  79,70,70,0,2,44,35,58,24,0,2,26,31,31,79,78,0,79,70,70,
  0,129,1,12,41,15,2,82,101,108,101,32,49,0,129,1,40,41,15,2,
  82,101,108,101,32,50,0,131,63,182,40,14,2,17,2,31,77,97,105,110,
  0,9 };

// структура определяет все переменные и события вашего интерфейса управления
struct {

    // input variables
    uint8_t switch_01; // =1 если переключатель включен и =0 если отключен
    uint8_t switch_02; // =1 если переключатель включен и =0 если отключен
    uint8_t switch_03; // =1 если переключатель включен и =0 если отключен

    // output variables
    char text_01[11]; // =строка UTF8 оканчивающаяся нулем
    char text_02[11]; // =строка UTF8 оканчивающаяся нулем
    char text_03[11]; // =строка UTF8 оканчивающаяся нулем
    uint8_t led_01_r; // =0..255 яркость красного цвета индикатора
    uint8_t led_01_g; // =0..255 яркость зеленого цвета индикатора
    uint8_t led_01_b; // =0..255 яркость зеленого цвета индикатора
    uint8_t led_03_r; // =0..255 яркость красного цвета индикатора
    uint8_t led_03_g; // =0..255 яркость зеленого цвета индикатора
```

```

uint8_t led_03_b; // =0..255 яркость зеленого цвета индикатора

// other variable
uint8_t connect_flag; // =1 if wire connected, else =0

} RemoteXY;
#pragma pack(pop)

//////////
//      END RemoteXY include      //
//////////

uint8_t switch_01_last = 2;
uint8_t switch_02_last = 2;
uint8_t switch_03_last = 2;

#define RX_PIN 4
#define TX_PIN 5
#define rel1_pin 14
#define rel2_pin 12
SoftwareSerial serial_UART(RX_PIN, TX_PIN);

float temperature;
int humidity;
int gas;
unsigned long climateCheak = -28000;
const long interval = 30000;
bool doorOpen = true;

// Методы
void parseCommand(String command);
void send_to_uart(String command, String param, String answer);
void led_door_rule(uint8_t Red, uint8_t Green, uint8_t Blue);
void led_system_rule(uint8_t Red, uint8_t Green, uint8_t Blue);
// Методы

void setup() {
    RemoteXY_Init ();
    // TODO you setup code
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(rel1_pin, OUTPUT);
    pinMode(rel2_pin, OUTPUT);
    Serial.begin(9600);
    serial_UART.begin(9600);
    send_to_uart("info","clim","");
}

void loop() {
    RemoteXY_Handler ();
    unsigned long currentTime = millis();

    if (currentTime - climateCheak >= interval){
        send_to_uart("info","clim","");
        climateCheak = millis();
    }
    if(switch_01_last != RemoteXY.switch_01){
        if(RemoteXY.switch_01 == 1){
            if (doorOpen){
                Serial.println("DOOR OPEN CANNOT PROTECTION");
                RemoteXY.switch_01 = 0;
            } else if (!doorOpen){
                digitalWrite(LED_BUILTIN, LOW);
                send_to_uart("guard","1","");
                //RemoteXY.led_01_g = 255;
                //send_to_uart("Led","on","");
            }
        }
    }
}

```

```

    }

    }else if(RemoteXY.switch_01 == 0){
        digitalWrite(LED_BUILTIN, HIGH);
        send_to_uart("guard","0","");
        led_system_rule(0,0,255);
    }
    switch_01_last = RemoteXY.switch_01;
}
if(switch_02_last != RemoteXY.switch_02){
    if(RemoteXY.switch_02 == 1){
        digitalWrite(rel1_pin, HIGH);
        Serial.println("Rele 1 - ON");
    } else if(RemoteXY.switch_02 == 0){
        digitalWrite(rel1_pin, LOW);
        Serial.println("Rele 1 - OFF");
    }
    switch_02_last = RemoteXY.switch_02;
}
if(switch_03_last != RemoteXY.switch_03){
    if(RemoteXY.switch_03 == 1){
        digitalWrite(rel2_pin, HIGH);
        Serial.println("Rele 2 - ON");
    } else if(RemoteXY.switch_03 == 0){
        digitalWrite(rel2_pin, LOW);
        Serial.println("Rele 2 - OFF");
    }
    switch_03_last = RemoteXY.switch_03;
}

if (serial_UART.available()) {
    String command = serial_UART.readStringUntil('\n'); // Читаємо рядок до символу нового рядка
    Serial.println(command);
    command.trim();
    parseCommand(command); // Викликаємо функцію для парсингу команд
}
}

void parseCommand(String command) {
    Serial.println("To_My:" + command);
    if (command.startsWith("info:")){
        int temIndex = command.indexOf("tem-");
        int humIndex = command.indexOf("hum-");
        int gasIndex = command.indexOf("gas-");
        int doorStateIndex = command.indexOf("door-");
        if (temIndex != -1 && humIndex != -1 && gasIndex != -1) {
            temperature = command.substring(temIndex + 4, command.indexOf(';', temIndex)).toFloat();
            humidity = command.substring(humIndex + 4, command.indexOf(';', humIndex)).toFloat();
            gas = command.substring(gasIndex + 4, command.indexOf(';', gasIndex)).toInt();
            doorOpen = command.substring(doorStateIndex + 5, command.indexOf(';', doorStateIndex)).toInt();
            dtostrf(temperature, 6, 2, RemoteXY.text_01);
            itoa(humidity, RemoteXY.text_02, 10);
            itoa(gas, RemoteXY.text_03, 10);
            if(doorOpen){
                led_door_rule(255,0,0);
            } else {
                led_door_rule(0,255,0);
            }
        }
    }
    }else if (command.startsWith("guard:")){
        if (command.substring(6) == "OK1") {
            Serial.println("System UNDER Protection");
            RemoteXY.switch_01 = 1;
            led_system_rule(0,255,0);
        } else if (command.substring(6) == "OK0") {

```

```

    Serial.println("System WITHOUT Protection");
    RemoteXY.switch_01 = 0;
    led_system_rule(0,0,255);
} else if (command.substring(6) == "Alarm") {
    Serial.println("Alarm");
    //RemoteXY.switch_01 = 0;
    led_system_rule(255,0,0);
}
} else if (command.startsWith("DoorState:")){
    if (command.substring(10) == "open") {
        Serial.println("Door OPEN");
        doorOpen = true;
        led_door_rule(255,0,0);
    } else if (command.substring(10) == "close") {
        Serial.println("Door CLOSE");
        doorOpen = false;
        led_door_rule(0,255,0);
    }
}
}

void send_to_uart(String command, String param, String answer){
    String command_to_send = command + ":" + param;
    serial_UART.println(command_to_send);
    Serial.println(command_to_send);
}

void led_door_rule(uint8_t Red, uint8_t Green, uint8_t Blue){
    RemoteXY.led_01_r = Red;
    RemoteXY.led_01_g = Green;
    RemoteXY.led_01_b = Blue;
}

void led_system_rule(uint8_t Red, uint8_t Green, uint8_t Blue){
    RemoteXY.led_03_r = Red;
    RemoteXY.led_03_g = Green;
    RemoteXY.led_03_b = Blue;
}

```

ДОДАТОК Г АКТ ВПРОВАДЖЕННЯ

ТОВ «Тірас-12»

21021, Україна

Вінницький р-н, Вінницька обл., місто Вінниця

пров. Хмельницького шосе 2, буд. 8

АКТ

ВПРОВАДЖЕННЯ

Цим актом підтверджується, що окремі результати магістерської кваліфікаційної роботи на тему: «Система охоронної сигналізації на основі платформи Arduino», яку виконав студент групи ІБС-23м Кутняк Максим за спеціальністю 125 «Кібербезпека та захист інформації», впроваджені в діяльність ТОВ «Тірас-12». Цей документ не є підставою для фінансових розрахунків.

Дата _____

Начальник відділу розробки охоронних приладів _____ Олександр ДЯЧУК

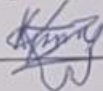


ДОДАТОК Д ІЛЮСТРАТИВНА ЧАСТИНА

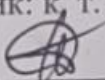
ІЛЮСТРАТИВНА ЧАСТИНА

«СИСТЕМА ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO»

Виконав: студент 2 курсу, групи ІБС-23м
спеціальності 125 Кібербезпека та захист
інформації

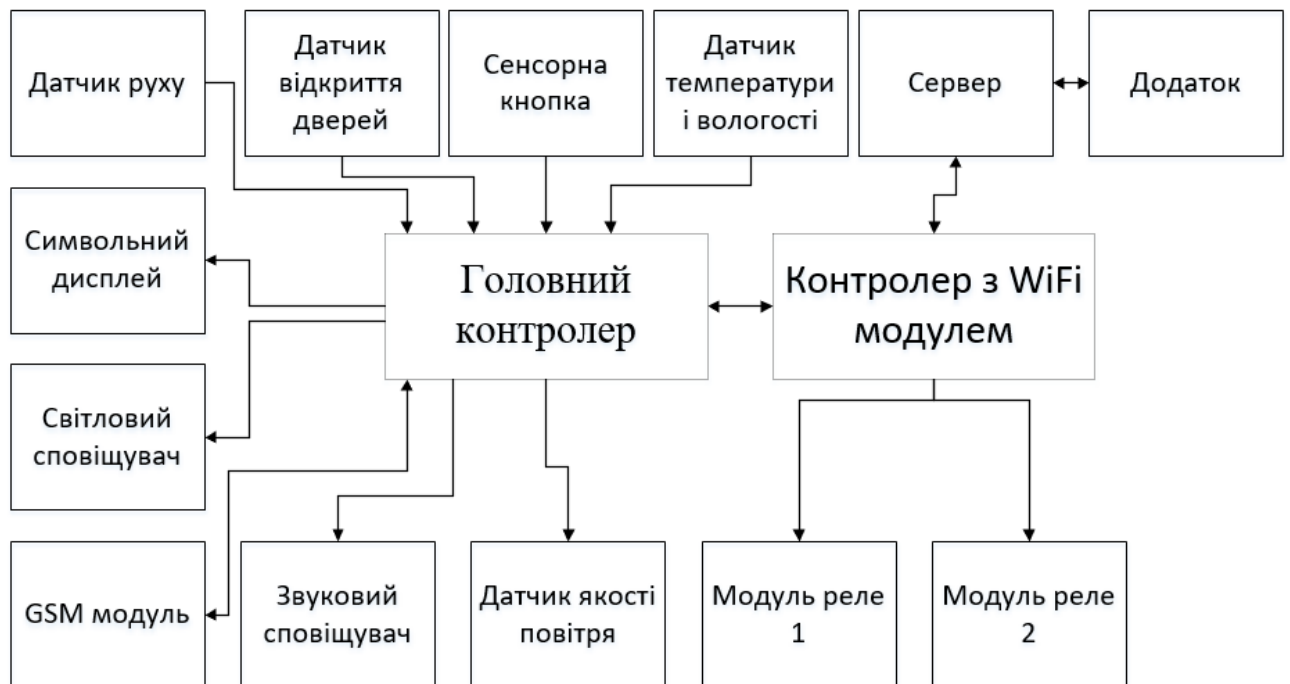
 Максим КУТНЯК

Керівник: к. т. н., доцент каф. ЗІ

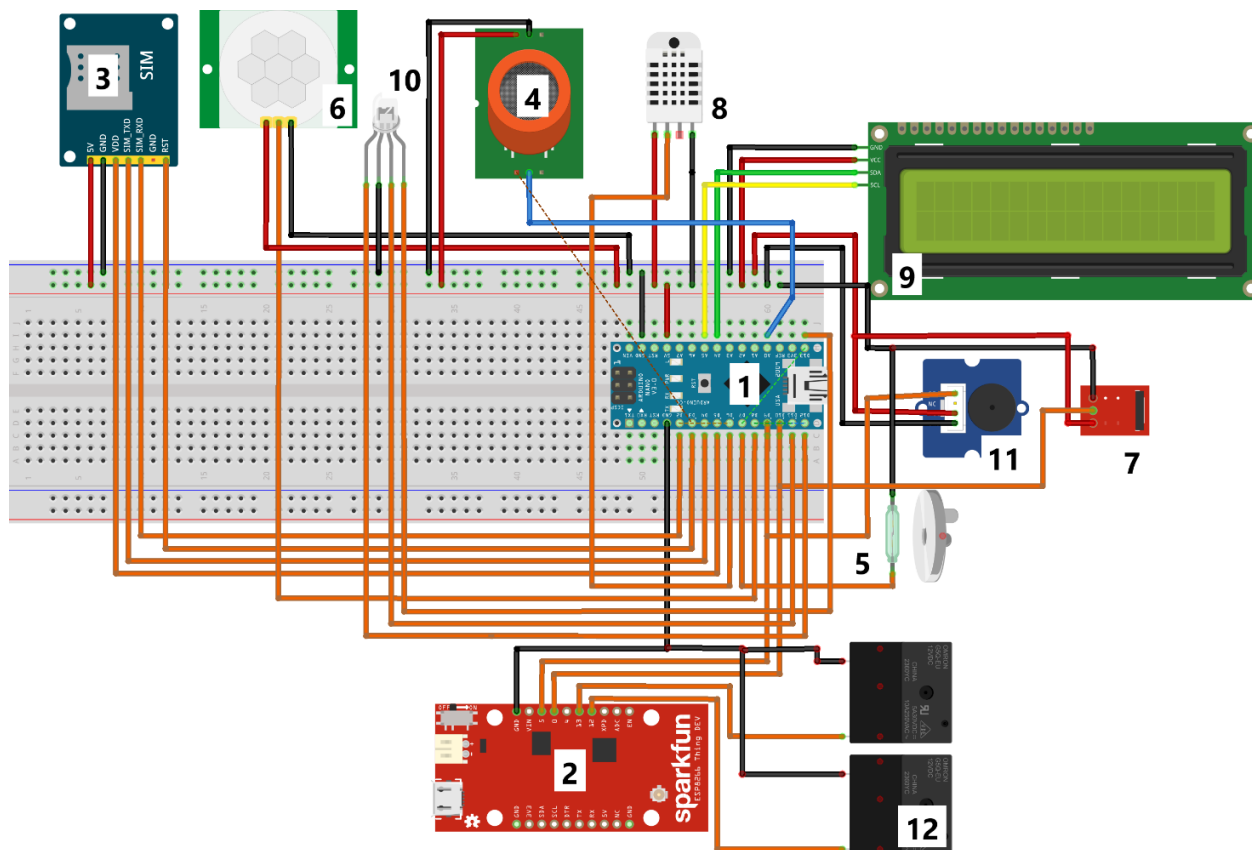
 Леонід КУПЕРШТЕЙН

«17» 12 2024 р.

ЗАГАЛЬНА СТРУКТУРНА СХЕМА СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO

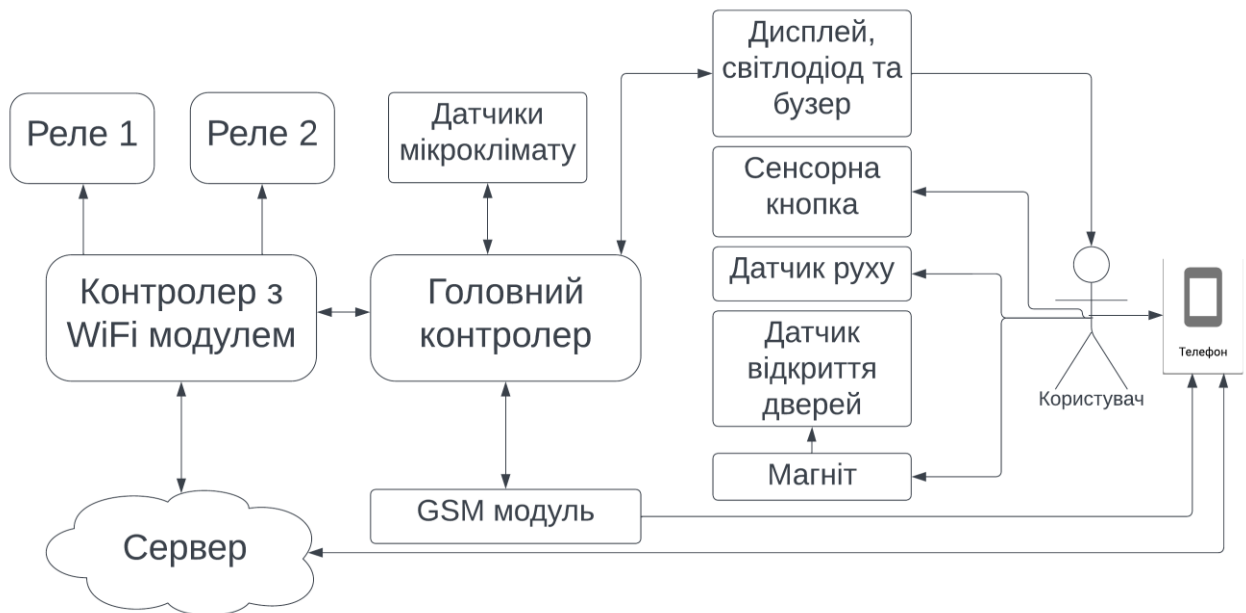


МОНТАЖНА СХЕМА СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ОСНОВІ ПЛАТФОРМИ ARDUINO



fritzing

UML ДІАГРАМА КОМУНІКАЦІЇ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ НА ПЛАТФОРМІ ARDUINO



UML ДІАГРАМА СТАНІВ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЙ НА ПЛАТФОРМИ ARDUINO

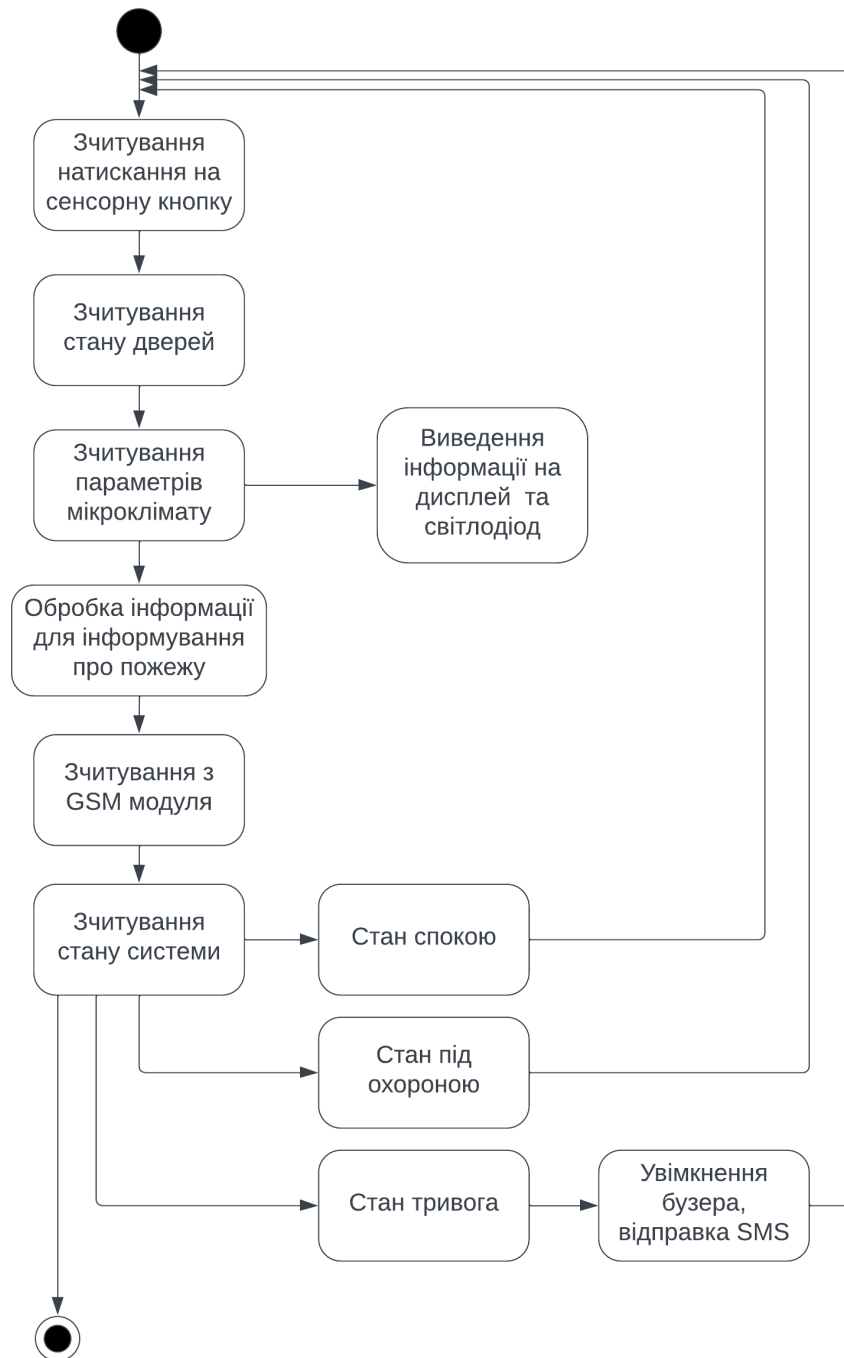


СХЕМА АЛГОРИТМУ РОБОТИ КОНТРОЛЕРА З WIFI

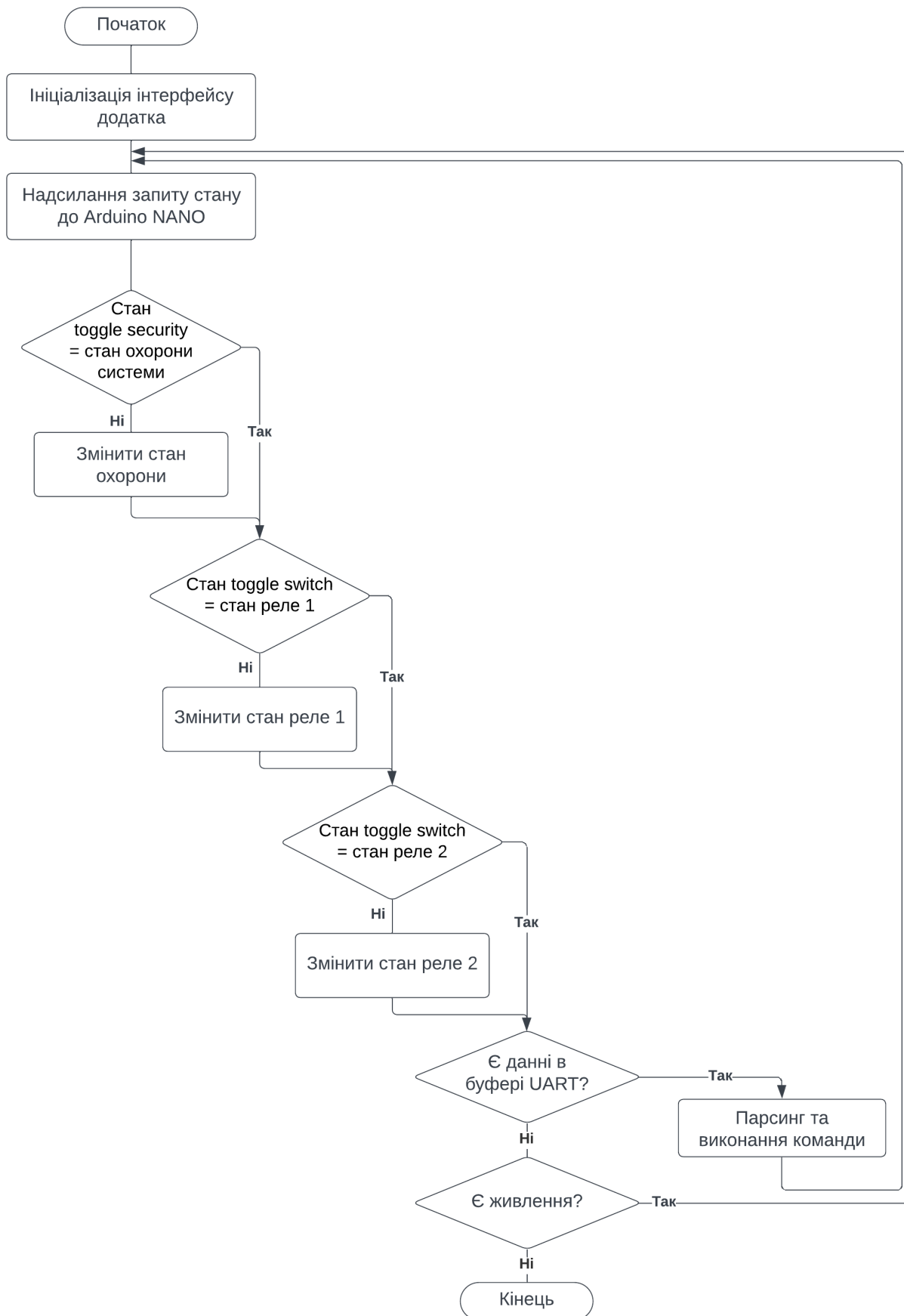
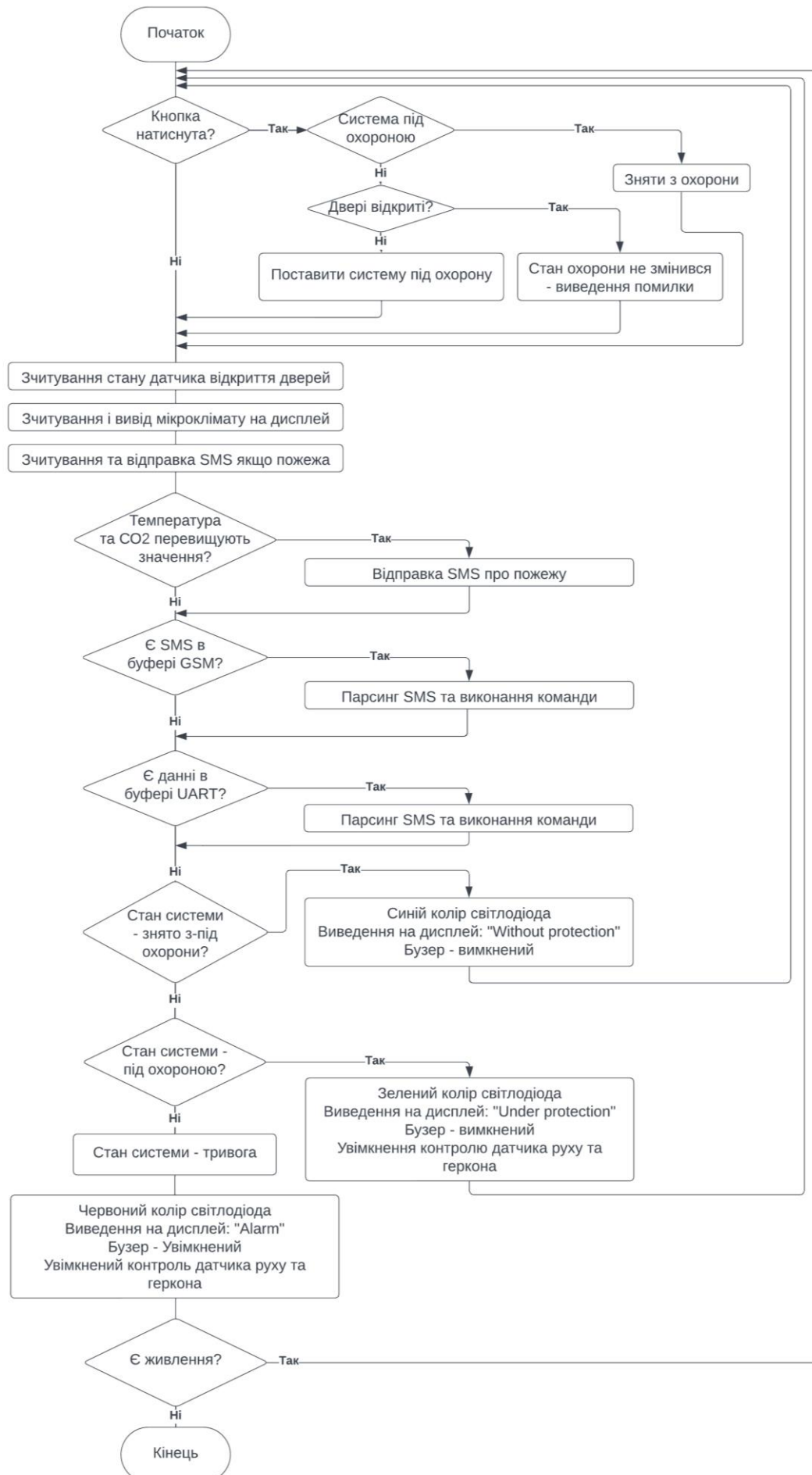
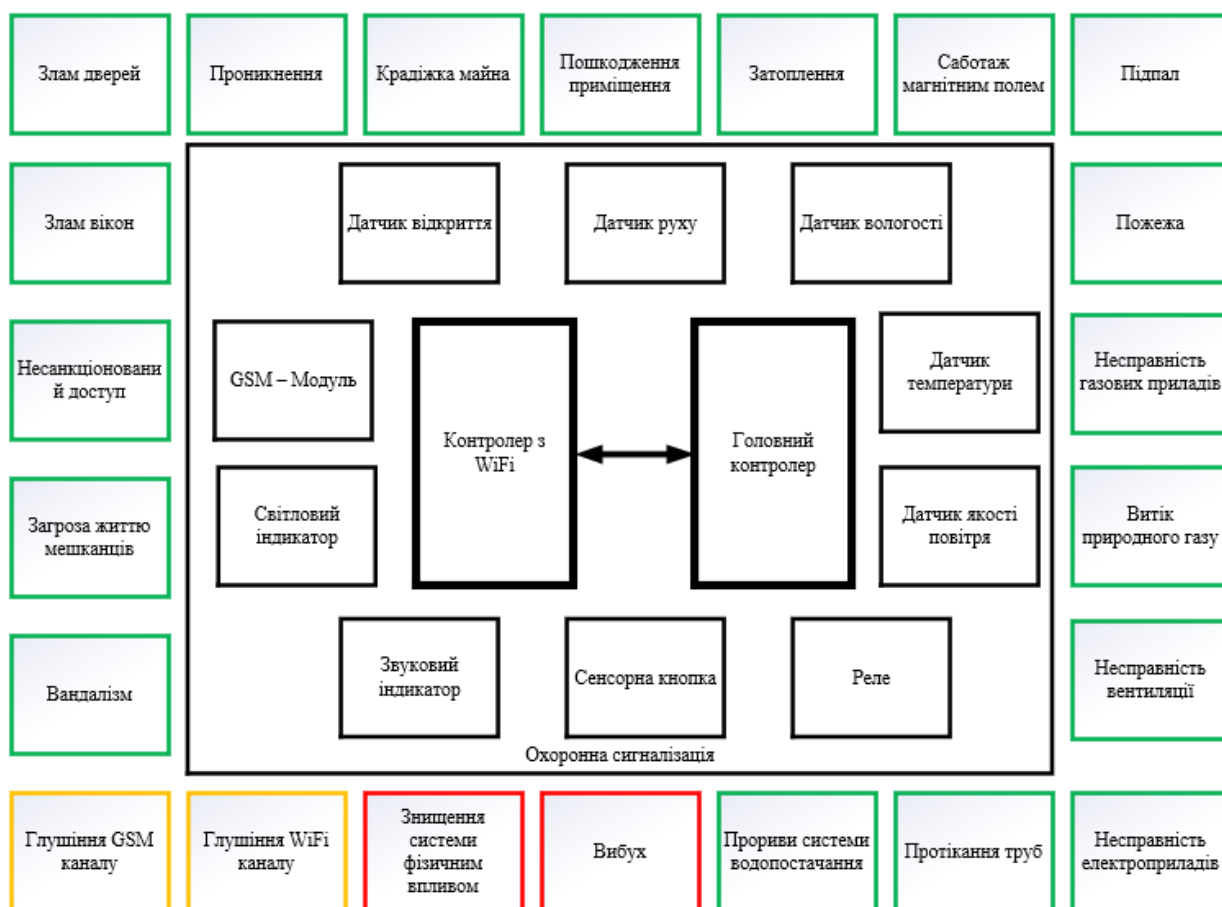


СХЕМА АЛГОРИТМУ РОБОТИ ГОЛОВНОГО КОНТРОЛЕРА



ПЕРЕЛІК ПОТЕНЦІЙНИХ ЗАГРОЗ, ЯКІ СИСТЕМА ОХОРОННОЇ СИГНАЛІЗАЦІЇ ПОКРИВАЄ



ВІДГУК

керівника магістерської кваліфікаційної роботи

студента (-ки) Кутняка Максима Миколайовича
(прізвище, ім'я, по батькові)

на тему «Система охоронної сигналізації на основі платформи Arduino»

Тема магістерської кваліфікаційної роботи є актуальною, оскільки питання забезпечення безпеки приміщень, моніторингу параметрів мікроклімату та інтеграції інтелектуальних систем безпеки є пріоритетними в сучасному суспільстві. Використання платформи Arduino для створення охоронної сигналізації дозволяє розробити економічно ефективне, масштабоване та просте в обслуговуванні рішення, що відповідає сучасним вимогам безпеки та зручності для користувачів.

Основні матеріали роботи, включно з пояснювальною запискою, ілюстративними матеріалами, структурними та монтажними схемами, UML-діаграмами та супровідною документацією, виконані на високому рівні. Вибір компонентів та датчиків (таких як датчики руху, газу, температури та вологості) обґрунтовано з врахуванням їх технічних можливостей та необхідних функціональних вимог. Поставлені у завданні задачі були повністю вирішені. Проект охоплює проектування апаратної частини системи та розробку програмного забезпечення для Arduino та проведення тестування системи.

Робота має значну практичну цінність, оскільки створена система охоронної сигналізації може бути використана для захисту приватних та комерційних об'єктів. Використання недорогих компонентів платформи Arduino дозволяє зменшити собівартість системи, що підвищує її конкурентоспроможність на ринку систем безпеки.

Ілюстративна частина виконана на високому рівні. Схеми, таблиці, UML-діаграми та графіки чіткі та інформативні. Пояснювальна записка написана логічно, з дотриманням наукового стилю, та відображає повний цикл розробки системи — від визначення вимог до тестування та оцінки результатів.

З огляду на вищевказане, вважаю, що магістерська кваліфікаційна робота заслуговує на оцінку за ECTS «А», а її автор – Кутняк М.М. – на присвоєння кваліфікації: ступінь вищої освіти магістр, галузь знань - "Інформаційні технології", спеціальність - "Кібербезпека та захист інформації", освітня програма - "Безпека інформаційних і комунікаційних систем".

Керівник магістерської
кваліфікаційної роботи

А.Г.А. проф. ЗІ
(посада, науковий ступінь, вчене звання)


(підпис)

Куперштейн Л.М.
(ініціали, прізвище)

ВІДГУК ОПОНЕНТА
на магістерську кваліфікаційну роботу
студента (-ки) Кутняка Максима Миколайовича

на тему: «Система охоронної сигналізації на основі платформи Arduino»

Тема магістерської кваліфікаційної роботи є актуальною, оскільки питання забезпечення безпеки приміщень та моніторингу параметрів мікроклімату набувають все більшого значення. Використання платформи Arduino для створення системи охоронної сигналізації дозволяє розробити економічно ефективне та адаптивне рішення, що відповідає сучасним вимогам користувачів. Робота повністю відповідає завданню, оскільки у процесі дослідження студент досяг поставлених цілей та реалізував визначені завдання.

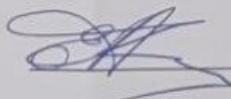
Основні матеріали роботи, включно з ілюстративною частиною, пояснювальною запискою та супровідною документацією, виконані на високому рівні. Використано сучасні методи аналізу та проектування, що дозволило створити оптимальну архітектуру системи охоронної сигналізації. Вибір компонентів та датчиків обґрунтований із врахуванням технічних можливостей та необхідних функціональних вимог. Структурні та монтажні схеми чітко відображають конфігурацію системи, а UML-діаграми забезпечують зрозумілу візуалізацію взаємодії компонентів та алгоритмів роботи.

Магістерська кваліфікаційна робота має високу практичну цінність. Система охоронної сигналізації, розроблена на основі платформи Arduino, може бути застосована у реальних умовах для захисту приватних та комерційних об'єктів. Також позитивною рисою є те, що в роботі реалізовано повний цикл розробки від аналізу вимог до реалізації та тестування системи.

Як недолік можна відмітити те, що не приділено уваги оптимізації енергоспоживання системи для забезпечення автономного режиму, а також розширенню кількості охоронних датчиків.

У цілому, вважаю, що магістерська кваліфікаційна робота виконана на високому рівні і заслуговує на оцінку за ECTS «А», а її автор – Кутняк М.М. – на присвоєння кваліфікації: ступінь вищої освіти магістр, галузь знань - "Інформаційні технології", спеціальність - "Кібербезпека та захист інформації", освітня програма - "Безпека інформаційних і комунікаційних систем".

Опонент
к. т. н., доц. каф. ПЗ



Олена КОВАЛЕНКО