МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
(повна назва випускної кафедри)

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

СИСТЕМА ДИСТАНЦІЙНОГО КЕРУВАННЯ СИСТЕМАМИ БЕЗПЕКИ БУДІВЕЛЬ ЗА ДОПОМОГОЮ МІКРОКОНТРОЛЕРІВ

Виконав:

студент групи 1КІ-23 зі спеціальності

123 – „Комп'ютерна інженерія"

Мазур Павло Юрійович

Науковий керівник:

к.т.н., доцент Розломій І.О.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент:

к.т.н., доцент кафедри інформаційної безпеки Черкаського національного університету ім. Б. Хмельницького - Т.А. Стабецька
(науковий ступінь, вчене звання, прізвище та ініціали)

Черкаси, 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

Кафедра *комп'ютерної інженерії та інформаційних технологій*

(повна назва випускної кафедри)

Спеціальність *123 – «Комп'ютерна інженерія»*

(шифр і назва спеціальності)

Освітня програма *«Комп'ютерна інженерія»*

(назва освітньої програми)

ЗАТВЕРДЖУЮ

Завідувач кафедри КІ та ІТ

_____ В. І. Хотунов

(підпис)

«___» _____ 202_ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТИ БАКАЛАВРА
Мазур Павло Юрійович
(прізвище, ім'я по батькові студента в називному відмінку)

1. Тема кваліфікаційної роботи бакалавра "Система дистанційного керування системами безпеки будівель за допомогою мікроконтролерів"

Науковий керівник роботи к.т.н. Розломій І.О.
(науковий ступінь, вчене звання, прізвище, ім'я по батькові)

Затверджено наказом ЧДБК від «__» _____ 202_ р. № ____

2. Строк подання студентом пояснювальної записки кваліфікаційної роботи бакалавра «__» _____ 202_ р.
3. Вихідні дані пояснювальної записки кваліфікаційної роботи бакалавра *охоронні системи безпеки з функцією сповіщення про загрози та дистанційним керуванням керуванням.*
4. Зміст пояснювальної записки кваліфікаційної роботи бакалавра (перелік питань, які потрібно розробити) *дешева охоронна система на базі мікроконтролера, постановка задачі роботи, принципова схема пристрою, розробка алгоритму та програмної частини пристрою, тестування пристрою та аналіз отриманих результатів*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Рисунок 2.9 – Функціональна схема пристрою.

5. Дата видачі завдання «__» _____ 202_ р.

КАЛЕНДАРНИЙ ПЛАН

№ пп	Назва етапів виконання пояснювальної записки кваліфікаційної роботи бакалавра	Терміни виконання етапів	Примітка про виконання
1	Вступ		
2	Пошук та аналіз літературних джерел		
3	Розділ 1 Елементна база для реалізації логічних елементів		
4	Розділ 2 Автоматичне виявлення та виправлення помилок при передачі інформації		
5	Розділ 3 Принципова схема блоку передаючої частини пристрою		
6	Висновки		
8	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра (чистовий варіант)		
9	Здача пояснювальної записки кваліфікаційної роботи бакалавра на кафедру для рецензування (за 14 днів до захисту)		
10	Перевірка пояснювальної записки кваліфікаційної роботи бакалавра на наявність ознак плагіату (за 10 днів до захисту)		

Студент _____ Мазур Паво Юрійович
(підпис) (прізвище, ім'я по батькові студента)

Науковий керівник _____ к.т.н., доцент Розломій І.О.
(підпис) (науковий ступінь, вчене звання, прізвище, ім'я по батькові)

АНОТАЦІЯ

У кваліфікаційній роботі бакалавра розроблено систему дистанційного керування системою безпеки будівлі на базі мікроконтролера Arduino з використанням GSM-зв’язку. Представлено структурну та принципову схеми пристрою, що забезпечує виявлення несанкціонованого проникнення та оперативне інформування користувача за допомогою SMS.

У роботі проаналізовано існуючі охоронні системи (Ajax, Risco), обґрунтовано вибір мікроконтролера Arduino Pro Mini, GSM-модуля SIM800C, RTC-модуля DS3231, сенсорів руху та інших компонентів. Описано алгоритми функціонування пристрою, реалізовані на мові C++ у вигляді окремих програмних модулів (скетчів), а також використання програмного забезпечення для конфігурації системи.

Розроблена система демонструє високу гнучкість, модульність та можливість розширення, що робить її ефективним рішенням для охорони малих приміщень, зокрема гаражів, майстерень, приватних будинків. Проведено тестування, що підтвердило працездатність пристрою в реальних умовах та наявність можливостей для подальшого вдосконалення.

Кваліфікаційна робота бакалавра містить 53 аркуші пояснювальної записки, 37 рисунків та 2 додатки.

Ключові слова: Arduino, GSM, охоронна система, мікроконтролер, сигналізація, SMS-сповіщення, сенсори, скетч, бездротовий зв’язок.

ANNOTATION

In this bachelor's qualification thesis, a remote building security control system based on the Arduino microcontroller with GSM communication was developed. The structural and circuit diagrams of the device are presented, which ensure the detection of unauthorized access and promptly notify the user via SMS.

The work includes an analysis of existing security systems (Ajax, Risco), justification for selecting the Arduino Pro Mini microcontroller, SIM800C GSM module, DS3231 RTC module, motion sensors, and other components. The system’s operation algorithms were implemented in C++ as separate software modules (sketches), along with the use of specialized software for system configuration.

The developed system demonstrates high flexibility, modularity, and expandability, making it an effective solution for securing small premises such as garages, workshops, and private houses. Testing confirmed the functionality of the device under real conditions and highlighted opportunities for further improvement.

The bachelor's qualification thesis consists of 53 pages of explanatory note, 37 figures, and 2 appendices.

Keywords: Arduino, GSM, security system, microcontroller, alarm, SMS notification, sensors, sketch, wireless communication.

Зміст

ВСТУП ... 7
РОЗДІЛ 1 ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ОХОРОННИХ СИСТЕМ СПОВІЩЕННЯ ... 9
1.1 Призначення та область застосування охоронної системи сповіщення ... 9
1.2 Огляд існуючих аналогів охоронних систем сповіщення ... 10
1.2.1 GSM-сигналізація Ajax ... 10
1.2.2 GSM-сигналізація Risco ... 16
1.3 Постановка задачі для розробки системи ... 22
РОЗДІЛ 2 ОПИС І ОБҐРУНТУВАННЯ ПРОЄКТНИХ РІШЕНЬ ОХОРОННОЇ СИСТЕМИ СПОВІЩЕННЯ ... 24
2.1 Обґрунтування вибору методів та інструментів розробки ... 24
2.2 Опис функціонування системи ... 28
2.3 Розробка структурної схеми системи ... 29
2.4 Розробка функціональної схеми системи ... 30
Висновки до розділу 2 ... 33
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ОХОРОННОЇ СИСТЕМИ СПОВІЩЕННЯ ... 34
3.1 Розробка блок-схем та опис алгоритмів функціонування системи ... 34
Висновки до розділу 3 ... 41
РОЗДІЛ 4 ВПРОВАДЖЕННЯ ОХОРОННОЇ СИСТЕМИ В ЕКСПЛУАТАЦІЮ ... 42
4.1 Налаштування системи ... 42
4.2 Тестування системи ... 43
Висновки до розділу 4 ... 45
ВИСНОВКИ ... 47
ДОДАТКИ ... 51
Додаток А ... 51
Додаток Б ... 53

ВСТУП

Забезпечення безпеки приватних об'єктів є надзвичайно актуальною проблемою, особливо в умовах зростання кількості крадіжок та несанкціонованих проникнень. Розвиток технологій дав змогу перейти до більш ефективних рішень – охоронних систем, які використовують мікроконтролери та датчики. Більшість сучасних професійних систем охорони є дорогими та складними у встановленні, що робить їх недоступними для пересічного користувача. Саме створення недорогих, надійних і функціональних охоронних систем на основі Arduino та GSM-технологій є актуальним напрямом, який сприяє поширенню доступної автоматизованої безпеки.

Метою даної кваліфікаційної роботи є розробка ефективної та економічно доцільної охоронної системи на базі мікроконтролера Arduino, здатної контролювати параметри датчиків і дистанційно повідомляти користувача про виявлені зміни за допомогою GSM-зв’язку.

Для досягнення поставленої мети необхідно реалізувати такі завдання:

1. Проаналізувати предметну область охоронних систем та систем сповіщення.
2. Здійснити огляд і порівняння існуючих рішень у сфері охоронних технологій.
3. Визначити функціональні та програмні вимоги до розроблюваної системи.
4. Спроєктувати охоронну систему з урахуванням необхідних датчиків та модулів.
5. Реалізувати систему на основі мікроконтролера Arduino з використанням модуля GSM.
6. Провести тестування створеної системи та оцінити її готовність до практичного використання.

Об’єктом дослідження є процес контролю змін параметрів у середовищі за допомогою датчиків охоронної системи.

Предмет дослідження є охоронна система на основі мікроконтролера, яка забезпечує дистанційне сповіщення користувача про зміну параметрів, виявлених датчиками.

У роботі використовуються методи порівняльного аналізу для вибору найбільш ефективних рішень, методи проєктування електронних схем, програмування мікроконтролерів Arduino, а також тестування системи в умовах, наближених до реального використання.

Джерелами інформації стали наукові статті, технічна документація на компоненти Arduino та GSM-модулі, інструкції з використання датчиків, аналітичні матеріали щодо сучасних рішень у сфері охоронних технологій.

Наукова новизна полягає у поєднанні доступного апаратного забезпечення з можливістю автономного контролю стану середовища та дистанційного сповіщення, що дозволяє адаптувати систему для використання в умовах обмеженого бюджету.

Розроблена охоронна система може бути використана для охорони приватних будинків, гаражів, офісів або невеликих об’єктів. Вона дозволяє оперативно реагувати на небезпеку та зменшити ймовірність несанкціонованого доступу, при цьому є недорогим і зручним рішенням для широкого кола користувачів.

РОЗДІЛ 1 ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ОХОРОННИХ СИСТЕМ СПОВІЩЕННЯ

1.1 Призначення та область застосування охоронної системи сповіщення

Охоронна система сповіщення в сучасному вигляді – це інтегрований технологічний комплекс, що використовує передові електронні компоненти, датчики високої точності та багаторівневі механізми захисту. Такі системи здатні ефективно охоплювати та контролювати значні площі об’єктів і забезпечувати оперативне реагування на загрози [1].

Попит на автоматизацію охоронних процесів постійно зростає, що обумовлює розвиток галузі та значні прибутки компаній, які займаються проєктуванням, виготовленням і впровадженням таких рішень. У минулому охорона об’єктів покладалась виключно на людський ресурс, що потребувало значних витрат на підготовку кадрів, екіпірування. Сьогодні ж автоматизація дозволяє мінімізувати людське втручання, зменшити витрати, усунути людський фактор і підвищити ефективність охорони.

Завдяки модульному підходу охоронні системи легко адаптуються до конкретних потреб об’єкта – від житлових будинків до великих виробничих комплексів. Усі компоненти системи об'єднуються через центральний контролер і можуть працювати автономно при виході з ладу окремого модуля, часто навіть повідомляючи користувача про несправність. Функціональні можливості таких систем досить широкі: автоматичне відкривання дверей, реагування на рух, дим, чадний газ, зміну освітлення, температури тощо [2].

Недоліком є висока вартість таких систем. Для охорони невеликих об'єктів (наприклад, гаража чи майстерні) використання повноцінної системи з десятками модулів є економічно необґрунтованим. У таких випадках доцільним є використання бюджетного рішення – GSM-сигналізації на базі мікроконтролера Arduino[3]. Вона забезпечує передачу повідомлень про вторгнення за допомогою стільникового зв’язку, має компактні розміри та базовий набір необхідних датчиків для моніторингу навколишнього середовища.

Живлення системи може здійснюватися як від мережі, так і від батарейок або акумуляторів, що забезпечує автономність. Завдяки модульності та доступності компонентів, система легко піддається модернізації та обслуговуванню. Arduino - платформа спрощує процес налаштування та програмування.

Серед недоліків такої системи варто зазначити обмежену сферу застосування та нижчу точність роботи датчиків у порівнянні з професійними охоронними системами. Проте для побутового використання це – оптимальне рішення за співвідношенням ціна/якість.

1.2 Огляд існуючих аналогів охоронних систем сповіщення

1.2.1 GSM-сигналізація Ajax

На сучасному ринку представлено велику кількість GSM-сигналізацій, які різняться між собою функціональністю, технічними характеристиками, типами датчиків та вартістю. Для визначення ключових вимог до розроблюваної системи доцільно здійснити аналіз наявних комерційних рішень, виявити їхні сильні сторони й обмеження. Це дозволить сформувати перелік необхідних функцій для подальшого проєктування власного пристрою.

Ajax Systems – український виробник систем безпеки, заснований у Києві [4]. Компанія відома своїми інноваційними технологіями, що дозволило їй стати лідером не лише на вітчизняному ринку, а й зайняти гідне місце серед європейських конкурентів. Наразі в Ajax працює понад 1500 співробітників у різних країнах світу, а сам бренд згадується у рейтингу «Forbes». Системи безпеки цього виробника мають великий попит, і їх можна побачити в більшості сучасних будівель. Завдяки широкому асортименту продукції, користувачі мають можливість легко підібрати охоронний комплекс з урахуванням індивідуальних потреб. У межах дослідження розглядатиметься класичні GSM-сигналізації Ajax, до складу якої входять магнітно-контактні датчики та сенсори руху [5] (рис. 1.1).

Рисунок 1.1 − GSM-сигналізація Ajax

Комплекс монтується безпосередньо на об'єкті, що охороняється. Усі елементи встановлюються в потрібних місцях та з’єднуються з центральним пристроєм (хабом) за допомогою бездротового зв’язку. Передача команд між датчиками здійснюється через радіоканал на декількох частотах, що ускладнює спроби глушіння сигналу. У разі виявлення проникнення система сповіщає власника або службу охорони (рис. 1.2).

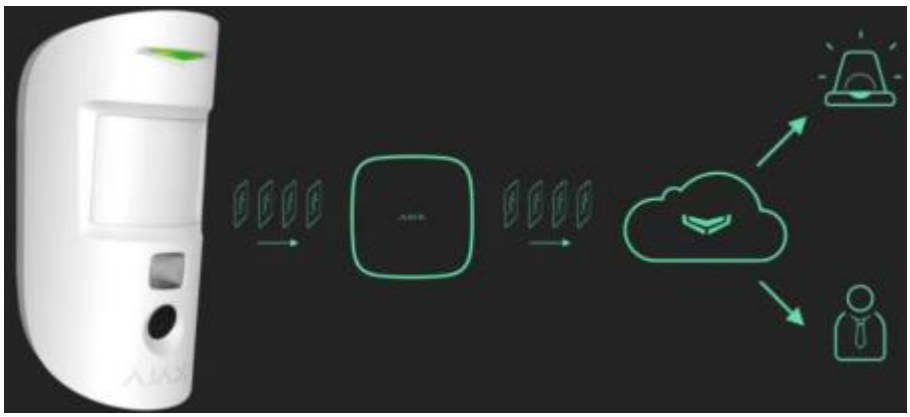

Рисунок 1.2 – Принцип процесу сповіщення системою Ajax[6]

Використання GSM-зв’язку забезпечує надійну передачу сигналу навіть у місцях, де інші канали, як-от Wi-Fi або Ethernet, можуть бути відсутні(рис. 1.3). А підключення додаткових каналів значно підвищує загальну стійкість системи, унеможливлюючи її повне блокування. Сигнал тривоги може надсилатися у вигляді SMS з текстовою інформацією або MMS із мультимедійними даними – фото, відео чи аудіо. Модульна структура системи дозволяє доповнювати її різними пристроями, які виконують додаткові функції моніторингу та охорони.

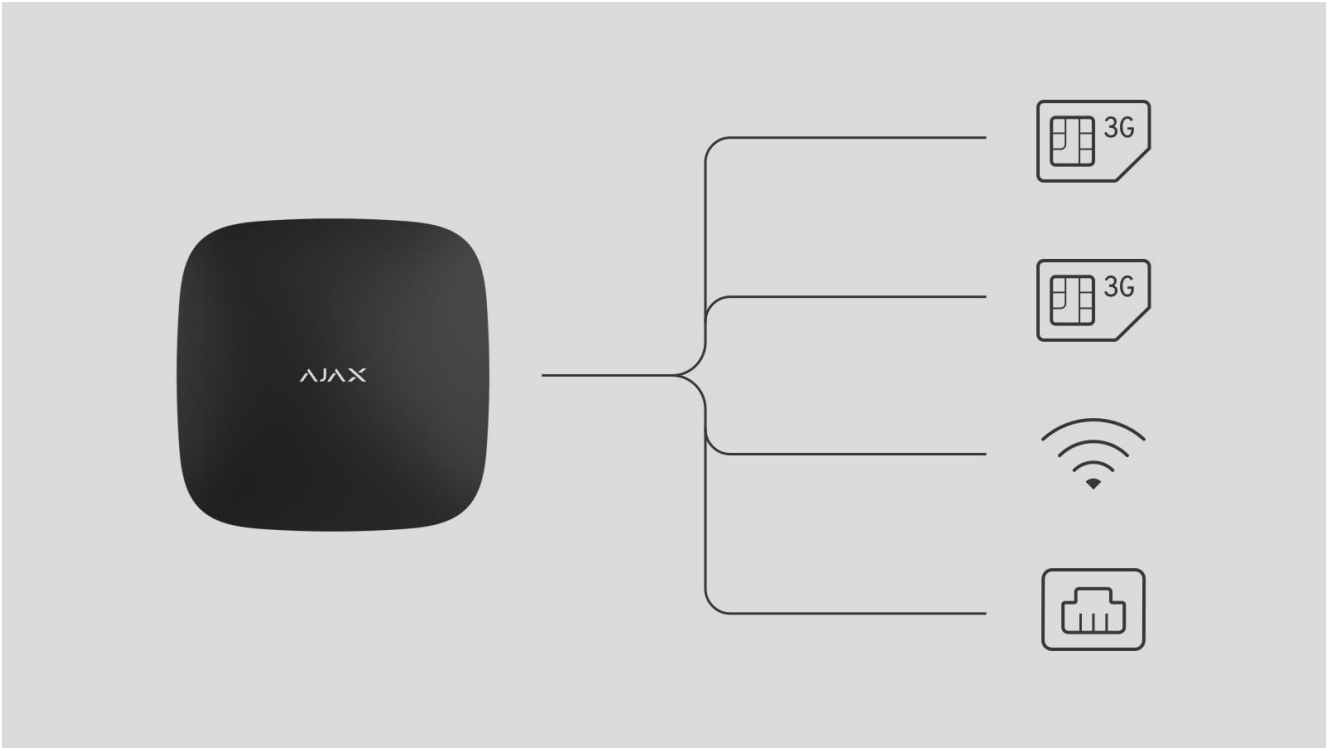

Рисунок 1.3 – Зображення каналів зв’язку

Одним із ключових елементів є датчик руху Ajax MotionProtect [7], що представлений у кількох варіаціях: інфрачервоні, мікрохвильові, ультразвукові, з використанням спектрального аналізу та функцією анти-маскування (рис. 1.4). Ці датчики здатні розрізняти людину та тварину, працюють у темряві, мають налаштування чутливості та можуть класифікувати об’єкти за зростом. MotionProtect входить до базової комплектації кожної охоронної системи Ajax.

Рисунок 1.4 – Датчик руху Ajax Motion Protect

Датчики розбиття скла Ajax GlassProtect [8] сприймають звуки, характерні для пошкодження скла (рис. 1.5). Вони застосовують двофакторний принцип аналізу сигналу, що знижує ймовірність хибних спрацьовувань. Такі датчики не реагують на побутові шуми, зокрема гавкіт собак або шум з вулиці. Максимальна відстань їх дії становить до 9 метрів.

Рисунок 1.5 – Датчик руху Ajax Glass Protect

Магнітно-контактні датчики Ajax [9] призначені для виявлення відкриття вікон і дверей (рис. 1.6). Вони є бездротовими та оснащуються високоякісними британськими герконами, які не залипають і витримують понад мільйон циклів роботи. Деякі моделі мають додаткову функцію виявлення зміни кута нахилу пристрою.

Рисунок 1.6 – Магнітно-контактний датчик Ajax

Пожежні датчики Ajax [10] забезпечують виявлення диму, підвищення температури, продуктів горіння та теплового випромінювання (рис. 1.7). Вони можуть мати вбудовану сирену, яка подає гучний сигнал при виникненні небезпеки, а також негайно надсилати повідомлення користувачеві.

Рисунок 1.7 – Пожежний датчик Ajax

Ajax LeaksProtect [11] – це датчик протікання води, який моментально виявляє появу вологи та надсилає сигнал тривоги на смартфон користувача (рис. 1.8). Після висихання води тривога автоматично скасовується. Корпус пристрою є водонепроникним (рівень захисту IP65), а батарея забезпечує автономну роботу до 5 років. Датчик дозволяє дистанційно контролювати свій стан і рівень заряду.

Рисунок 1.8 – Датчик протікання води Ajax

Системи Ajax здатні передавати тривожні сигнали в таких форматах як SMS-повідомлення – текстові сповіщення про тривогу, та MMS-повідомлення – мультимедійна інформація для пульта охорони.

Залежно від характеристик об’єкта й типу сигналізації, застосовуються комбіновані методи передачі сигналу. До системи можуть додатково інтегруватися пристрої для моніторингу споживання енергії. Наприклад, розумні розетки допомагають уникнути перегріву або займання у разі стрибків напруги, а світлодіодне підсвічування може слугувати індикатором енергоспоживання.

1.2.2 GSM-сигналізація Risco

Risco Group – це міжнародна компанія, що спеціалізується на розробці та виробництві інтегрованих рішень у сфері безпеки [12]. Компанія зарекомендувала себе як один із провідних гравців ринку завдяки надійності, функціональності та сучасному підходу до реалізації охоронних технологій. Risco пропонує широкий спектр охоронних систем для приватного, комерційного та промислового використання. Одним із основних напрямів розвитку компанії є GSM-сигналізації, які дозволяють організувати ефективну систему охорони з використанням мобільного зв’язку.

Risco Agility 4 – це флагманська система безпеки, яка об’єднує в собі передові функції охорони, моніторингу та керування (рис. 1.13) [13]. Вона підтримує двосторонній голосовий зв’язок, що дозволяє не лише отримувати сповіщення, а й спілкуватися з оператором охорони або особами, які перебувають у зоні контролю. Наявність вбудованих Wi-Fi і GSM-модулів гарантує стабільну передачу даних у режимі реального часу. Вбудований GPS-модуль надає можливість відстеження місцезнаходження пристрою, що особливо актуально для мобільних або віддалених об’єктів. Також система підтримує підключення відеокамер, що дає змогу реалізувати відеопідтримку тривоги – перегляд фото- або відеоінформації безпосередньо під час інциденту.

Рисунок 1.9 – GSM система Risco Agility 4

Risco LightSYS 2 – це гнучке охоронне рішення, яке однаково добре підходить для житлових і комерційних приміщень (рис. 1.14) [14]. Система оснащена GSM-комунікатором, що дозволяє відправляти сигнали тривоги за допомогою SMS, голосових дзвінків або GPRS. Завдяки модульному підходу її легко адаптувати до конкретних умов експлуатації.

Рисунок 1.10 – GSM система Risco LightSYS 2

Risco ProSYS Plus – професійна охоронна платформа, яка підтримує розширений набір функцій, включаючи дистанційне керування системою, резервне

живлення, багатоканальне повідомлення та інтеграцію з хмарними сервісами (рис. 1.15) [15]. Вона ідеально підходить для великих об'єктів, де потрібна багаторівнева безпека та масштабованість.

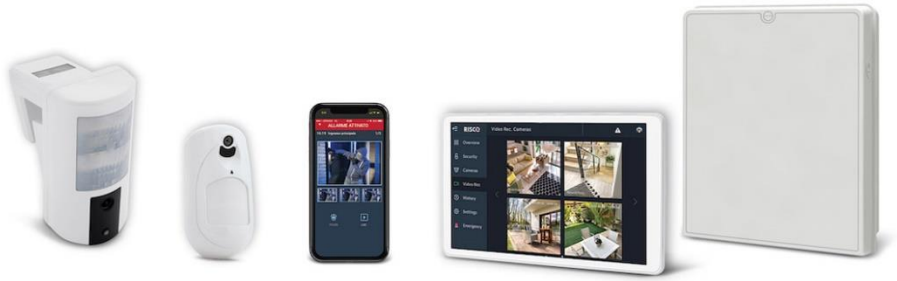

Рисунок 1.11 – GSM система Risco ProSYS Plus

Охоронні комплекси Risco комплектуються різноманітними бездротовими датчиками, які забезпечують виявлення широкого спектру загроз:

Risco RK150DTG3 – пасивний інфрачервоний сенсор руху (рис. 1.16), який вирізняється високою точністю та стійкістю до хибних спрацювань[16]. Завдяки цифровій обробці сигналу та вбудованому температурному сенсору, він автоматично коригує свою чутливість відповідно до змін температури в приміщенні. Його енергозберігаючий режим дозволяє працювати на одній батареї до 5 років, що значно знижує потребу в обслуговуванні.

Рисунок 1.12 – Датчик руху Risco RK150DTG3

Risco MC-311D – бездротовий магнітно-контактний сенсор для контролю відкриття дверей і вікон (рис. 1.17). Має компактні розміри, простий у встановленні та працює на відстані до 500 м від центрального блоку, що робить його ідеальним для великих площ.

Рисунок 1.13 – Датчик відкриття Risco MC-311D

Risco GD-542 – сенсор витоку газу, що здатен запобігти вибухонебезпечним ситуаціям шляхом оперативного виявлення парів газу в повітрі (рис. 1.18). Його робоча відстань становить до 100 м, а бездротовий зв’язок дозволяє гнучко обирати місце встановлення.

Рисунок 1.14 – Датчик газу Risco GD-542

Risco RK6F000 – детектор протікання води, який може бути встановлений у санвузлах, кухнях або технічних приміщеннях (рис. 1.19) [17]. Виявляє навіть незначні сліди вологи, надсилає тривогу й автоматично скидає її після зникнення загрози. Має дальність дії до 100 м.

Рисунок 1.15 – Датчик протікання Risco RK6F000

Wireless LuNAR FireWatch Smoke – бездротовий датчик диму, здатний виявити продукти горіння навіть на ранніх стадіях (рис. 1.20) [18]. Його можна встановити у житлових приміщеннях, офісах або складських зонах. Дальність зв’язку – до 500 м.

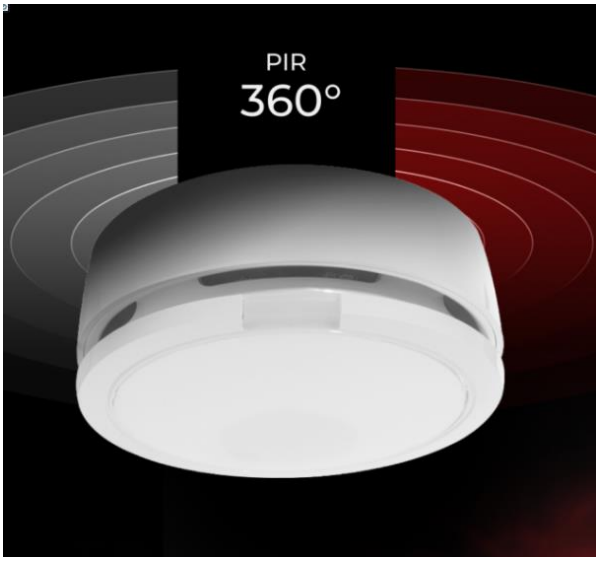

Рисунок 1.16 – Датчик диму Risco Wireless LuNAR FireWatch Smoke

Risco Glass Break Detectors – розбиття скла, який дозволяє своєчасно виявити спробу проникнення через вікно або скляну перегородку (рис. 1.21) [19]. Завдяки

широкому діапазону дії та точній обробці сигналу він забезпечує високий рівень надійності.

Рисунок 1.17 – Датчик розбиття скла Glass Break Detectors

Переваги використання GSM-систем Risco:

- Бездротова архітектура спрощує монтаж і зменшує втручання в інтер’єр.
- Широкий асортимент сумісних датчиків дозволяє адаптувати систему до будь-яких потреб.
- Можливість віддаленого керування та налаштування через мобільний додаток або веб-інтерфейс.
- Гнучке комбінування каналів зв’язку (GSM, Wi-Fi, Ethernet) підвищує стійкість до саботажу.

– Наявність резервного живлення гарантує безперебійну роботу під час відключення електроенергії.

У підсумку, GSM-системи Risco – це надійне рішення, яке поєднує високий рівень безпеки, гнучкість, простоту у використанні та інноваційні технології. Завдяки широким можливостям розширення та налаштування, ці системи підходять як для побутового, так і для професійного застосування.

1.3 Постановка задачі для розробки системи

У зв’язку з ускладненням безпекової ситуації в країні, значним зростанням рівня злочинності, зокрема крадіжок і несанкціонованих проникнень, розробка GSM-охоронної системи на базі Arduino є надзвичайно актуальною. Комерційні охоронні системи мають високу вартість, тому їх встановлення в невеликих приміщеннях часто є економічно невиправданим.

Запропонована система може знайти застосування у різних сферах: побутовій, офісній, транспортній та інших. Її перевагою є простота, компактність, енергоефективність та відносно низька собівартість. Використання GSM-зв’язку забезпечує дистанційне інформування користувача про потенційні загрози, що робить систему універсальною.

За результатами аналізу аналогічних систем і доступних рішень сформульовано основні завдання розробки:

1. Спроєктувати та зібрати основні апаратні модулі охоронної системи.
2. Розробити програмне забезпечення для керування пристроєм і забезпечення стабільної роботи.
3. Провести тестування, налаштування та налагодження системи відповідно до заданих технічних вимог.

Таким чином, реалізація цієї системи має практичну значущість і перспективи використання як у побуті, так і в малому бізнесі. Вдало виконана робота може стати цінним внеском у розвиток доступних технологій безпеки.

Висновки до розділу 1

У цьому розділі було здійснено глибокий аналіз предметної області та доцільність створення GSM-охоронної системи на основі Arduino. Показано, що для охорони невеликих об’єктів (наприклад, гаражів або майстерень) використання дорогих систем з численними модулями є недоцільним. Оптимальним рішенням є компактний пристрій із базовим функціоналом та низькою вартістю.

Проведено огляд існуючих аналогів, зокрема систем Ajax, Xiaomi та Risco, із порівнянням їхніх характеристик, датчиків і функціональних можливостей. Також обґрунтовано вибір компонентів і програмних засобів для реалізації проєкту. Визначено перелік інструментів, необхідних для розробки, та поставлено основні задачі.

Отримані результати підтверджують доцільність подальшої роботи над створенням ефективної, недорогої охоронної системи з можливістю віддаленого моніторингу, що задовольнить актуальні потреби ринку.

РОЗДІЛ 2 ОПИС І ОБҐРУНТУВАННЯ ПРОЄКТНИХ РІШЕНЬ ОХОРОННОЇ СИСТЕМИ СПОВІЩЕННЯ

2.1 Обґрунтування вибору методів та інструментів розробки

Відповідно до теми кваліфікаційної роботи, передбачено створення охоронної системи на основі плати мікроконтролера Arduino. Було обрано модель Arduino Pro Mini[20], головною перевагою якої є енергоефективність та стійкість до коливань живлення. Залежно від напруги живлення (3,3 В або 5 В), тактова частота пристрою становить відповідно 8 МГц або 16 МГц. Arduino Pro Mini (рис. 2.1) побудована на основі мікроконтролера ATmega328P, вирізняється компактними розмірами, відсутністю USB-інтерфейсу та мінімальним енергоспоживанням, що робить її ідеальним вибором для автономних мікропроєктів (Додаток А). Однак для завантаження програмного коду необхідно використовувати зовнішній програматор.

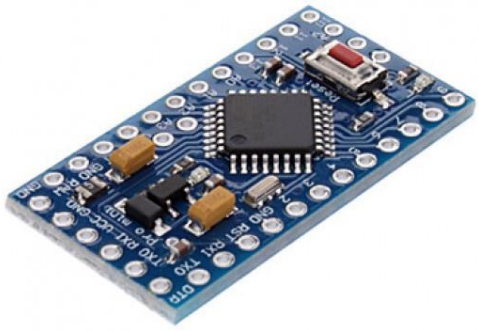

Рисунок 2.1 – Мікроконтролер Arduino Pro Mini

Для виявлення руху використовується сенсор MH-SR602 MINI (рис. 2.2) – невеликий, чутливий та енергоощадний датчик, який дозволяє фіксувати рух на відстані до 5 метрів [21]. Серед його основних переваг:

- низька вартість порівняно з аналогами;
- мінімальне енергоспоживання, що ідеально підходить для пристроїв на батарейному живленні;

– сумісність із більшістю мікроконтролерів (робоча напруга 3–5 В).

Рисунок 2.2 – Сенсор руху MH-SR602 MINI

Для забезпечення бездротового зв’язку використовується GSM-модуль SIM800C(L) [22] (рис. 2.3), який підтримує передачу SMS, дзвінки та передачу даних через GPRS. Його переваги:

– глобальна сумісність з мобільними мережами;
– підтримка передачі даних на сервери через GPRS;
– легка інтеграція з мікроконтролерами через UART-інтерфейс;
– низька вартість у порівнянні з іншими GSM-модулями.

Рисунок 2.3 – GSM-модуль SIM800C(L)

Для забезпечення стабільної роботи системи у випадку зависання мікроконтролера чи GSM-модуля, застосовується RTC-модуль DS3231 (рис. 2.4) [23] – годинник реального часу, який дозволяє точно відслідковувати час. Його особливості:

- висока точність завдяки термокомпенсованому кварцовому генератору;
- збереження даних після відключення живлення;
- підключення через I2C-інтерфейс;
- низьке енергоспоживання.

Рисунок 2.4 – Модуль годинника DS3231 RTC

Оскільки Arduino Pro Mini не має вбудованого USB-інтерфейсу, для програмування використовується перехідник CP2102 USB–UART TTL (рис. 2.5) [24]. Цей адаптер забезпечує:

- надійну конвертацію сигналів з USB у UART TTL;
- підтримку різних операційних систем (Windows, Linux, MacOS);
- просте підключення та високі швидкості передачі даних (до 2 Мбіт/с);
- роботу з напругами 3,3–5 В.

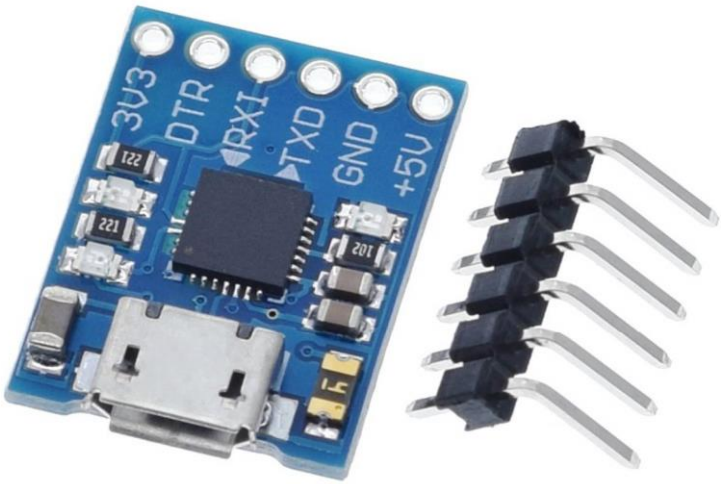

Рисунок 2.5– Модуль CP2102

Для написання програмного забезпечення обрано Arduino IDE – офіційне середовище розробки для мікроконтролерів Arduino (рис. 2.6) [25]. Серед її переваг:

- простий та інтуїтивно зрозумілий інтерфейс;
- наявність великої кількості бібліотек та прикладів;
- підтримка широкого спектра мікроконтролерів;
- можливість налагодження коду та відкритий вихідний код;
- гнучкість і розширюваність через додаткові плагіни.

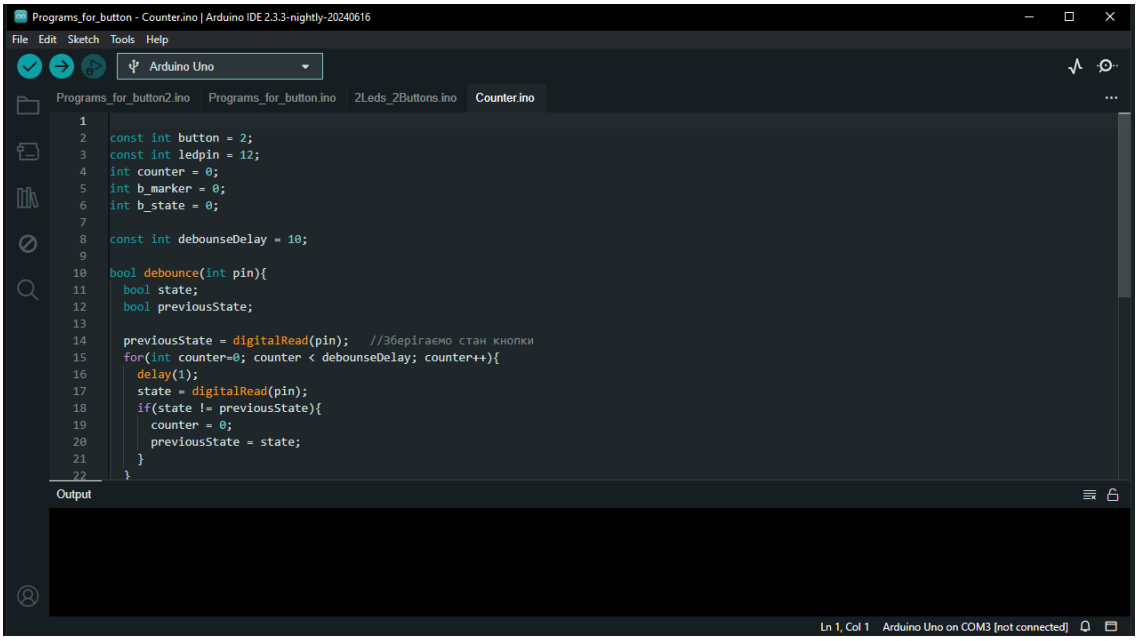

Рисунок 2.6 – Інтерфейс Arduino IDE

2.2 Опис функціонування системи

Охоронна система сповіщення реалізована у вигляді комплексу взаємопов’язаних модулів. До її складу входять основні компоненти: датчики, головний контролер (Arduino) та GSM-комунікаційний модуль. Кожен датчик виконує окрему функцію – виявлення руху, відкриття дверей, появи диму, витоку газу тощо. Всі датчики підключені до головного блоку керування, який обробляє вхідні сигнали та реагує відповідно до заданих умов.

У разі спрацювання датчика, сигнал надсилається до мікроконтролера. Система аналізує отриману інформацію, і якщо виявлено загрозу, активує відповідні дії: вмикає сирену та надсилає повідомлення користувачу через GSM-модуль. Повідомлення можуть передаватися у вигляді SMS або телефонного дзвінка. Для цього використовується бібліотека для Arduino, яка дозволяє взаємодіяти з GSM-модулем через AT-команди.

Функціональність системи можна розширити шляхом додавання додаткових сенсорів (наприклад, пожежних, газових), а також елементів дистанційного керування, таких як пульт або мобільний застосунок.

На рис. 2.7 представлено блок-схему функціонування системи: датчики фіксують зміни в середовищі, передають сигнали до мікроконтролера, який ініціює передачу повідомлень через GSM-модуль, сповіщення користувача.

Рисунок 2.7 – Діаграма функціонування системи

2.3 Розробка структурної схеми системи

Структурна схема системи сигналізації на основі Arduino зображена на рис. 2.8. Вона містить основні компоненти, які забезпечують повний цикл охоронної дії.

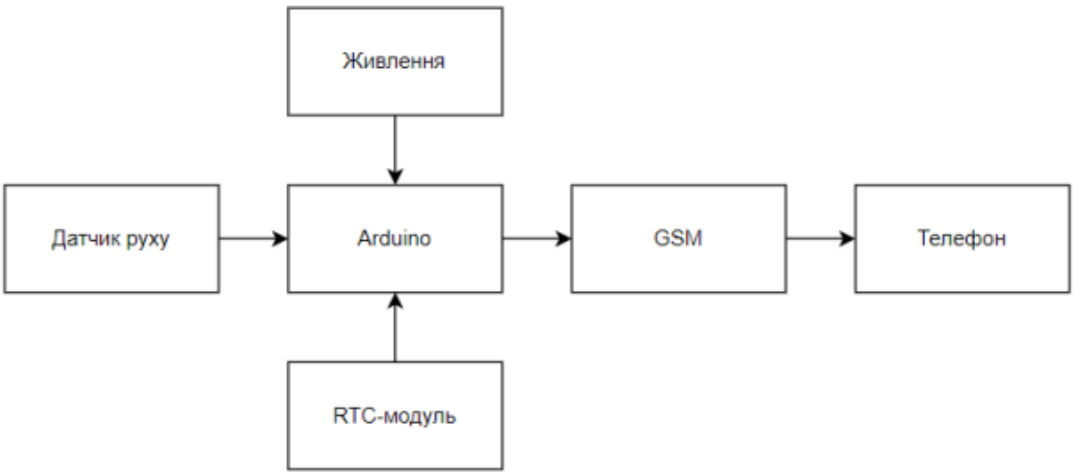

Рисунок 2.8 – Структурна схема системи

На першому рівні знаходяться сенсори (руху, відкриття, диму тощо), які виявляють зміну стану середовища. Сигнали з них надходять до мікроконтролера Arduino Pro Mini, який обробляє інформацію та приймає рішення щодо подальших дій. До Arduino також підключений RTC-модуль (DS3231), який забезпечує точне відслідковування часу та дозволяє налаштовувати інтервали для сповіщень.

GSM-модуль (SIM800C(L)) відповідає за зв'язок із мобільною мережею та надсилання тривожних повідомлень. Система також включає SIM-карту для реалізації зв’язку. Живлення забезпечується за допомогою батарейок типу AA або акумулятора, який можна заряджати від стандартного джерела живлення.

Завдяки структурі, система є автономною, гнучкою в налаштуванні та здатною забезпечити охорону невеликих приміщень різного призначення.

2.4 Розробка функціональної схеми системи

Розробка функціональної схеми для GSM-сигналізації на базі Arduino дозволяє чітко визначити основні компоненти системи, принцип їх роботи та взаємодії. Це дозволяє краще зрозуміти функціональні можливості системи, уникнути помилок у розробці, а також ефективно розподілити ресурси та час реалізації проєкту. На рис. 2.9 зображено функціональну схему охоронної системи, в якій показано основні елементи системи та їх взаємозв’язки. Також, на її основі, розроблено друковану плату(див. Додаток Б)

Для підключення датчика руху MH-SR602 MINI до плати Arduino Pro Mini необхідно:

- Провід VCC з’єднати з піном 5V на Arduino.
- Провід GND з’єднати з GND на Arduino.
- Провід OUT підключити до будь-якого цифрового входу, наприклад, до піну D6.

Після підключення можна перейти до програмування датчика: обробка сигналів та передача інформації через GSM-модуль.

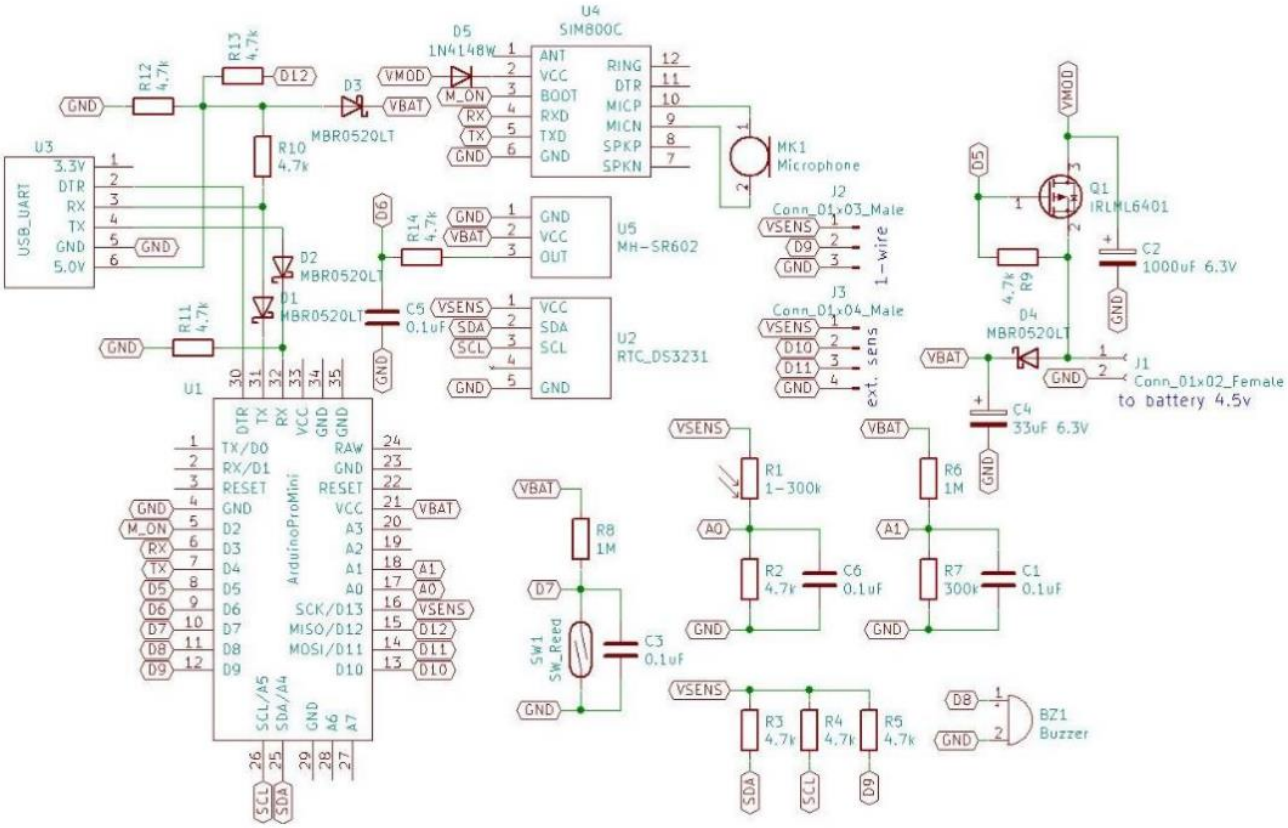

Рисунок 2.9 – Функціональна схема пристрою

Для підключення GSM-модуля SIM800C(L) до плати Arduino Pro Mini потрібно:

- Підключити живлення модуля в межах 3.4–4.4 В, а також GND.
- З’єднати TX модуля з RX Arduino, а RX модуля з TX Arduino.
- Підключити DTR модуля до RESET Arduino – це забезпечить автоматичне перезавантаження плати після передачі даних.
- Провід VBAT модуля під’єднати до A0 Arduino – це дозволяє контролювати рівень заряду.

Після фізичного підключення GSM-модуля необхідно встановити параметри зв’язку – швидкість передачі даних, формат повідомлень, номер SIM-карти тощо. Також рекомендується провести тестові виклики або відправлення SMS.

Підключення RTC-модуля DS3231 до плати Arduino виконується через інтерфейс I2C:

- Лінії SDA та SCL з’єднуються відповідно з відповідними пінками на Arduino.
- Також підключаються VCC та GND.
- Для роботи з RTC-модулем використовується бібліотека Wire.h, яка дозволяє взаємодіяти з пристроєм, зчитувати поточний час і дату та встановлювати значення за потреби.

Для завантаження програми в мікроконтролер Arduino Pro Mini використовується конвертер USB–UART CP2102. Підключення здійснюється за схемою:

- VCC модуля CP2102 → VCC Arduino;
- GND → GND;
- TX → RXI;
- RX → TXO;
- DTR → DTR.

Цей модуль дозволяє комп’ютеру взаємодіяти з Arduino для програмування або налагодження системи. Підключення CP2102 є необхідною частиною розробки при використанні Arduino Pro Mini, оскільки плата не має власного USB-інтерфейсу.

Висновки до розділу 2

У даному розділі було детально розглянуто проєктні рішення, що стосуються розробки охоронної системи сповіщення. Система включає кілька основних модулів: сенсори, центральний блок керування та комунікаційну мережу. Сенсори фіксують зміни в навколишньому середовищі та передають інформацію до центрального блоку, який обробляє отримані дані та виконує необхідні дії — наприклад, надсилає повідомлення або активує сигнал тривоги.

Для передачі повідомлень у системі використовується GSM-модуль, який дозволяє надсилати SMS та здійснювати дзвінки. Можливості системи можуть бути розширені шляхом додавання нових сенсорів, таких як датчики диму, полум’я або витоку газу, а також підключення додаткових пристроїв для дистанційного керування.

Усі елементи системи — зокрема, датчики руху, Arduino, модулі RTC і GSM, а також адаптер CP2102 MICRO USB to UART TTL — з’єднуються між собою за допомогою дротів і роз’ємів, що забезпечує надійний обмін сигналами та даними.

У розділі були представлені структурна та функціональна схеми системи, які наочно ілюструють її компоненти, логіку роботи та взаємодію між елементами.

Загалом, запропоновані інженерні рішення демонструють ефективність системи на базі Arduino для охорони різних об’єктів. Представлена розробка може слугувати базою для подальшого удосконалення та розширення функціональних можливостей охоронної системи.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ОХОРОННОЇ СИСТЕМИ СПОВІЩЕННЯ

3.1 Розробка блок-схем та опис алгоритмів функціонування системи

Наступним етапом створення системи є розробка скетч-файлів. Скетчі (від англ. sketch) – це програмні файли, призначені для написання коду мікроконтролерів, зокрема для платформи Arduino. Вони містять інструкції, що визначають логіку взаємодії мікроконтролера з периферійними пристроями, такими як датчики, модулі, дисплеї тощо. Розробка скетчу базується на мові програмування C++, яка після компіляції трансформується у машинний код, зрозумілий для мікроконтролерів. Використання скетчів дозволяє ефективно реалізовувати програмне забезпечення для проєктів, що передбачають інтеграцію з апаратними компонентами, а також проводити тестування функціональності системи та експериментувати з різними електронними елементами.

Для завантаження скетчів на мікроконтролер та його модулі застосовується утиліта xLoader. Це спеціалізований інструмент, який дає змогу передавати скомпільований програмний код на плату Arduino без використання інтегрованого середовища розробки (IDE). Головна перевага xLoader полягає в можливості інсталювати код на мікроконтролер навіть за відсутності доступу до Arduino IDE або без його встановлення на комп’ютер.

Процедура завантаження коду за допомогою xLoader відрізняється простотою: користувач обирає скомпільований файл, визначає COM-порт для зв’язку з платою, вказує тип мікроконтролера та швидкість передачі даних. Після цього програма автоматично передає дані через обраний інтерфейс. Додатково xLoader дозволяє інсталювати завантажувач (bootloader) на плату Arduino, що є критичним у випадках пошкодження або відсутності вбудованого завантажувача.

xLoader є безкоштовним програмним забезпеченням з відкритим вихідним кодом, що забезпечує гнучкість у його модифікації під конкретні потреби. Утиліта підтримує роботу з ОС Windows, Linux та macOS та доступна для завантаження на офіційному ресурсі розробників.

Розглянемо ключові скетчі системи та їх базову логіку. Перший із них – **EeRingQueue**, який реалізує шаблонний клас для роботи з кільцевою чергою у EEPROM. Кільцева черга – структура даних, де нові елементи записуються поверх найстаріших при заповненні буфера, забезпечуючи циклічне використання пам'яті.

Клас **EeRingQueue** включає методи:

- **put()** – додавання елемента;
- **get()** – отримання елемента;
- **update()** – оновлення існуючого елемента;
- **reset()** – скидання черги;
- **size()** та **capacity()** – отримання поточної кількості елементів та максимальної ємності.

Стан черги зберігається у структурі **state**, що містить:

- **tail** – індекс наступного елемента для додавання;
- **size** – кількість активних елементів. Додатково клас зберігає адресу в EEPROM для розміщення стану черги та її даних, що забезпечує стійкість до перезавантажень.

Другий ключовий скетч – **Config**, який керує налаштуваннями системи. У ньому визначено:

- Структуру **SchEntry** для опису подій розкладу (дні тижня, тип події, дія, час початку та завершення).
- Методи перевірки коректності налаштувань (**validate**) та отримання дій за подіями (**getAction**).

Клас **ConfigSerial** розширює функціонал **Config**, надаючи методи:

- **read()/write()** – зчитування та запис налаштувань у EEPROM;
- **toBase64()/fromBase64()** – серіалізація/десеріалізація даних у формат BASE64.

Збереження налаштувань у EEPROM гарантує їхню стійкість під час відключення живлення. Це критично для GSM-сигналізації на базі Arduino, де конфігурація має залишатися незмінною після перезапусків. Обидва скетчі забезпечують модульність, гнучкість та надійність роботи системи.

На рисунку 3.1 представлено блок-схему, яка ілюструє базовий алгоритм роботи скетчу Config.

Рисунок 3.1 – Блок-схема алгоритму файлу Config

Далі було розроблено скетч-бібліотеку Event, яка визначає структуру даних Event для фіксації інформації про події в системі. Структура включає час події у форматі Unix, тип події (EventNumber), значення (val), дії (Action), що виконуються при настанні події.

Бібліотека надає методи для серіалізації даних у формати CSV, текстовий та інші, а також містить шаблони рядків для форматування виводу. Основне призначення – централізоване управління подіями та відповідними реакціями, що критично для систем моніторингу, безпеки тощо.

Наступним буде розглянутий скетч Hardware. Скетч призначений для керування пристроєм, що здійснює моніторинг стану об'єкта та параметрів

навколишнього середовища. В основі його архітектури лежать два взаємопов'язані класи, які забезпечують повноцінну функціональність системи.

Клас Hardware відповідає за безпосередню взаємодію з апаратними компонентами системи. Він реалізує функції зчитування показників з різних датчиків, включаючи вимірювання температури, рівня освітленості, стану дверних контактів та датчиків руху. Особливу увагу приділено управлінню електроживленням - клас містить методи для перевірки джерела живлення, вмикання та вимикання харчування обладнання. Для інформування користувача реалізовано функціонал генерації звукових сигналів різної тривалості. Важливою складовою є робота з системним часом, для чого використовуються спеціалізовані бібліотеки RTClib.h та Wire.h.

Паралельно з основним класом працює TempHistory, який спеціалізується на зберіганні та аналізі історичних даних про зміни температури. Це дозволяє відстежувати динаміку змін температурного режиму та виявляти аномалії.

Архітектура класу Hardware передбачає використання спеціальних структур даних:

- tempStat для накопичення статистичної інформації про температурні показники;
- sensors як контейнера для зберігання всіх даних, отриманих від підключених датчиків.

Серед ключових методів класу варто відзначити init() для ініціалізації апаратних ресурсів, isExtPower() для визначення джерела живлення, pwrOn() та pwrOff() для керування електропостачанням. Функція sensors() забезпечує доступ до актуальних показників датчиків, тоді як now() і setTime() відповідають за роботу з системним часом. Для генерації звукових сигналів використовуються методи beep() та beep(int), а оновлення інформації про стан батареї здійснюється через updButVoltage().

Слід зазначити, що використання статичних методів для роботи з датчиками, з одного боку, спрощує процес інтеграції нових компонентів, але з іншого - може ускладнити адаптацію системи до змін у конфігурації обладнання. Аналогічно,

застосування статичної змінної tempHistory, незважаючи на зручність централізованого доступу до температурних даних, може негативно впливати на використання оперативної пам'яті.

Загальна реалізація скетчу Hardware демонструє високу ефективність при інтеграції різноманітних сенсорів та периферійних пристроїв. Використання спеціалізованих бібліотек для роботи з інтерфейсом I2C та системним часом значно спрощує процес синхронізації та обміну даними між окремими компонентами системи, забезпечуючи при цьому необхідну гнучкість і масштабованість рішення.

Розглядаючи детальніше функціонал скетчу Hardware, варто відзначити його багаторівневу структуру. На початковому етапі відбувається ініціалізація всіх системних компонентів: підключаються необхідні бібліотеки, визначаються константи та конфігуруються апаратні параметри. Особлива увага приділяється налаштуванню пінів мікроконтролера, які відповідають за взаємодію з периферійними пристроями.

Після успішної ініціалізації система переходить до основного циклу роботи. На цьому етапі постійно відбувається моніторинг стану всіх підключених датчиків. Дані про температуру, освітленість, стан дверей та активність руху зчитуються з заданою періодичністю. Одночасно система аналізує джерело живлення, що дозволяє їй адаптувати робочий режим до поточних умов.

Важливим аспектом роботи є накопичення та обробка температурних даних. Система веде детальну статистику, фіксуючи не лише поточні показники, а й аналізуючи їх динаміку. Для цього реалізовано спеціальний механізм збереження історичних даних, який активується кожні дві години. Це дозволяє відстежувати довгострокові зміни мікроклімату.

Реакція на події реалізована через систему звукових сповіщень. Генерація сигналів відбувається при виявленні будь-якої нестандартної ситуації: відкритті дверей, фіксації руху чи спрацьовуванні додаткових сенсорів. Це забезпечує своєчасне сповіщення користувача про потенційно небезпечні ситуації.

Завершальним етапом кожного циклу роботи є процедура управління енергоспоживанням. Система аналізує поточні потреби та при необхідності

переводить окремі компоненти в енергоефективний режим. Це дозволяє значно продовжити автономну роботу пристрою без втрати його функціональності.

Така організація робочого процесу забезпечує високу стабільність системи та її здатність ефективно виконувати поставлені завдання в різних умовах експлуатації. Модульний принцип побудови додатково спрощує процес майбутнього оновлення або розширення функціоналу.

Скетч Reporter забезпечує взаємодію з модемом SIM800 для організації SMS-сповіщень та голосового зв'язку. В основі його роботи лежить бібліотека TinyGSMExtra, яка спрощує керування GSM-модемом та обробку помилок зв'язку.

Клас Reporter ініціалізується з параметрами COM-порта та цільового телефонного номера, на який будуть надсилатись повідомлення та здійснюватись виклики. Для роботи з модемом використовуються жорстко задані піни підключення – BOOT_PIN та MODEM_POWER_PIN, що забезпечує стабільність з'єднання.

Основним функціоналом класу є два ключові методи. Метод send() відповідає за формування та відправку текстових повідомлень. Він автоматично ініціалізує модем, перевіряє наявність мережі та виконує трансляцію повідомлення, при цьому підтримуючи міжнародне кодування через спеціальний макрос GF. У разі виникнення проблем метод повертає відповідний код помилки.

Метод call() реалізує механізм голосового виклику. Після стандартної ініціалізації модему він активує відображення номера, ініціює дзвінок на вказаний номер та підтримує зв'язок протягом стандартного інтервалу у 3 хвилини. Для коректного завершення роботи передбачено метод pwrDwnModem(), який гарантовано вимикає модем без втрати даних.

Особливу увагу приділено обробці помилок. Система розрізняє кілька типів проблем: від помилок ініціалізації та реєстрації в мережі до специфічних проблем під час відправки повідомлень чи здійснення дзвінків. Для зручності налагодження передбачено спеціальні макроси DBG, які дозволяють відстежувати стан системи в реальному часі.

Архітектура класу дозволяє легко інтегрувати його в існуючі системи моніторингу, забезпечуючи надійний канал зв'язку для надсилання сповіщень та екстрених викликів. Використання шаблонного підходу при формуванні повідомлень додатково розширює можливості персоналізації контенту SMS.

Клас **Scheduler** призначений для організації планування та відправки системних повідомлень через зовнішній модем. Він інтегрує кілька ключових компонентів:

Основні елементи класу:

- **DateTime** – для роботи з часовими мітками
- **Status** – зберігає дані про стан системи
- **Event** – містить інформацію про події
- **Reporter** – відповідає за відправку повідомлень

Клас містить ряд важливих полів, які забезпечують його функціональність. Поле **nextTry** (тип DateTime) визначає час наступної спроби відправки повідомлення, а **attempt** відстежує кількість вже виконаних спроб. Прапорець **reportStatus** вказує, чи потрібно надіслати звіт про стан системи, а структура **dailyStatus** фіксує час щоденного звіту та його статус (відправлено / не відправлено).

Для отримання актуальних даних клас використовує посилання на об'єкт **Status**, який містить інформацію про поточний стан системи, та **Reporter**, що відповідає за комунікацію через модем. Конструктор **Scheduler** приймає об'єкт **Config**, з якого отримує налаштування щоденного звіту (**dailyReportTime**).

Функціональність класу охоплює автоматичне планування повторних спроб відправки у разі невдачі, контроль періодичності звітів та взаємодію з GSM-модулем. Він забезпечує стабільну роботу системи сповіщень, адаптуючись до змін у конфігурації та стані мережі. Використання **Config** для налаштувань дозволяє легко змінювати параметри роботи без модифікації коду.

Таким чином, **Scheduler** виступає ключовою ланкою у системі управління подіями, забезпечуючи своєчасне інформування та гнучкість у роботі з повідомленнями.

Висновки до розділу 3

У цьому розділі було здійснено проєктування блок-схем і детальний опис алгоритмів роботи розробленої системи. Розглянуто ключові етапи створення системи, серед яких – встановлення зв’язку між мікроконтролером та GSM-модулем, розробка програмного забезпечення для обробки інформації, а також реалізація алгоритмів оповіщення у випадку фіксації небажаних подій.

Окрема увага приділялась процесу написання скетчів для мікроконтролерів на базі платформи Arduino. Ці скетчі, що є програмним кодом на мові C++, дозволяють мікроконтролеру взаємодіяти з електронними компонентами системи. Для завантаження коду на плату без застосування стандартного середовища Arduino IDE була використана програма xLoader, яка відзначається простотою у користуванні та зручністю для користувачів, яким не потрібно встановлювати додаткове програмне забезпечення.

У розділі також було проаналізовано декілька прикладів скетчів та пояснено основні принципи їх роботи. Результатом проведеної роботи стало створення логічної структури та алгоритмічного забезпечення, здатного забезпечити швидке реагування системи на надзвичайні ситуації, тим самим підвищуючи рівень безпеки користувачів і їхнього майна. Отже, можна зробити висновок, що ефективне функціонування охоронної системи залежить від якісного проєктування апаратної частини, належного програмного забезпечення та оптимальних алгоритмів реагування на загрози.

РОЗДІЛ 4 ВПРОВАДЖЕННЯ ОХОРОННОЇ СИСТЕМИ В ЕКСПЛУАТАЦІЮ

4.1 Налаштування системи

Для економії часу, для налаштування системи був використаний застосунок GSMguard (рис. 4.1) [26]. Він значно спрощує процес налаштувань у порівнянні з прямим редагуванням скетчів. Додаток пропонує інтуїтивно зрозумілий інтерфейс для керування всіма критичними параметрами системи.

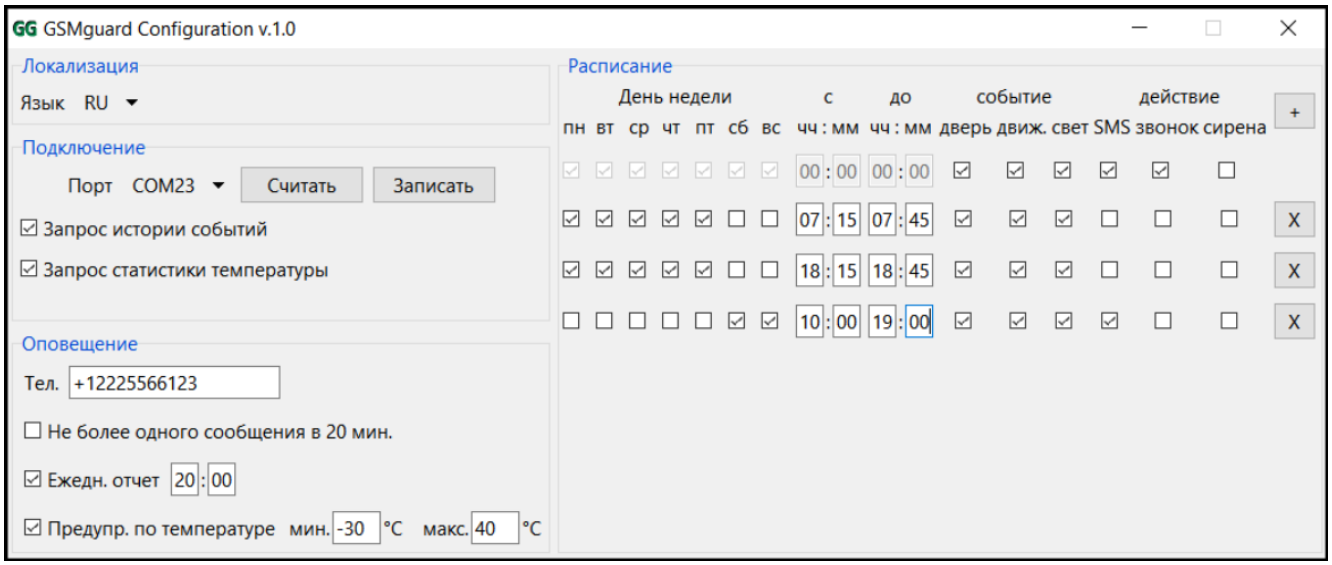

Рисунок 4.1. – Інтерфейс GSMguard

Ключові функції додатку включають:

1. Конфігурування основних параметрів:
   - Встановлення та зміна телефонного номера для отримання сповіщень
   - Налаштування періодичності та часу щоденних звітів
   - Визначення температурних меж для спрацювання тривоги
2. Візуалізація даних:
   - Побудова детальних графіків температурних коливань за місяць
   - Відображення історії подій у зручному для аналізу вигляді
3. Моніторинг стану системи:
   - Перегляд логів з усіма зареєстрованими подіями(рис. 4.2)
   - Аналіз роботи датчиків у реальному часі

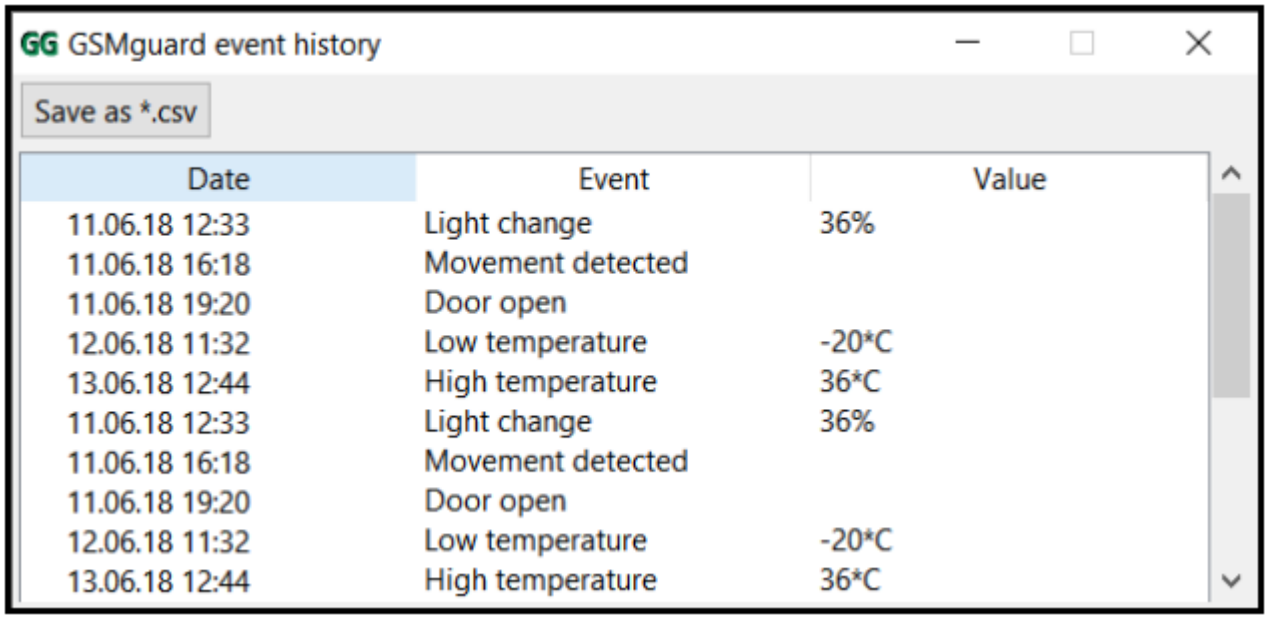

Рисунок 4.2 – Вікно подій

4.2 Тестування системи

Під час тестування пристрою були перевірені наступні функціональні можливості:

– після збирання та прошивки пристрій потребує початкового налаштування конфігурації через відповідну утиліту;

– при підключенні до USB-порту нормальна робота системи призупиняється, а черга непріоритетних повідомлень автоматично очищується;

– у разі невдалої спроби надіслати SMS, пристрій здійснює повторні спроби за таким інтервалом: через 2, 5, 10, 20 хвилин, двічі по 40 хвилин, а далі – кожні 12 годин;

– звуковий сигнал активується на 30 секунд; спрацьовування датчиків таких подій, як «відкриття дверей», «рух» і «зміна освітлення», обмежено інтервалом у 20 хвилин, щоб уникнути надмірного реагування;

– якщо живлення пристрою відсутнє понад 3 години, історія температурних вимірювань автоматично анулюється.

Проведене тестування підтвердило коректну роботу системи. Пристрій було змонтовано на дверях гаража. Для інтенсивного випробування функція розкладу не

застосовувалась – кожна подія фіксувалась шляхом надсилання SMS. У середньому надходило близько 5 SMS на день: по дві при заїзді та виїзді (датчики відкриття дверей і освітлення), а також одне щоденне повідомлення. На даний момент живлення забезпечується трьома батарейками типу AA, які видають напругу 4,1 В.

У ході експлуатації було виявлено, що при зниженні температури RIP-датчик може подавати хибні сигнали. У зв’язку з цим у майбутньому доцільно замінити його на більш стабільну модель, адаптовану до екстремальних температур. Аналогічна ситуація спостерігається з температурним датчиком – при показниках нижче -10 °C він демонструє некоректні значення. Тому рекомендовано використовувати його більш термостійкий аналог.

Охоронний модуль має компактну конструкцію (рис. 4.3), що дозволяє легко встановити його у малопомітному місці. Якщо використовується герконове реле, пристрій можна без зайвих труднощів інтегрувати біля дверей або вікна (рис. 4.4).

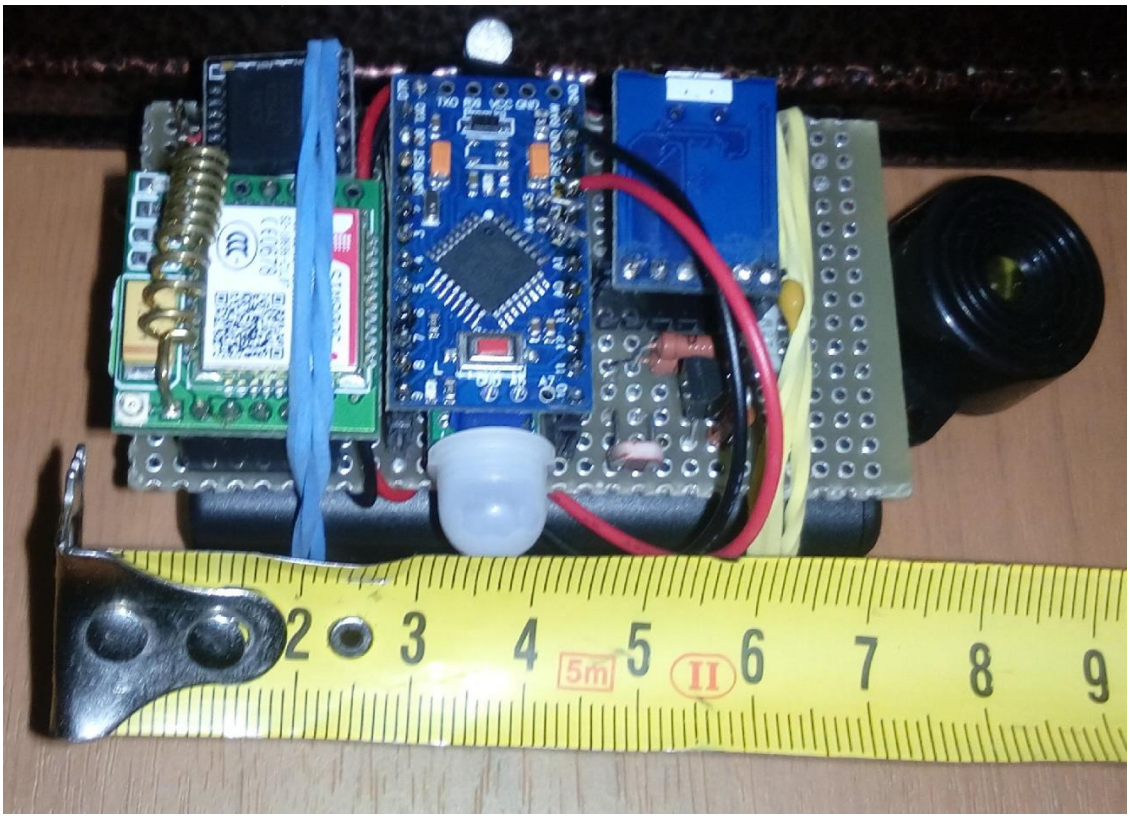

Рисунок 4.3 – Розміри охоронної системи

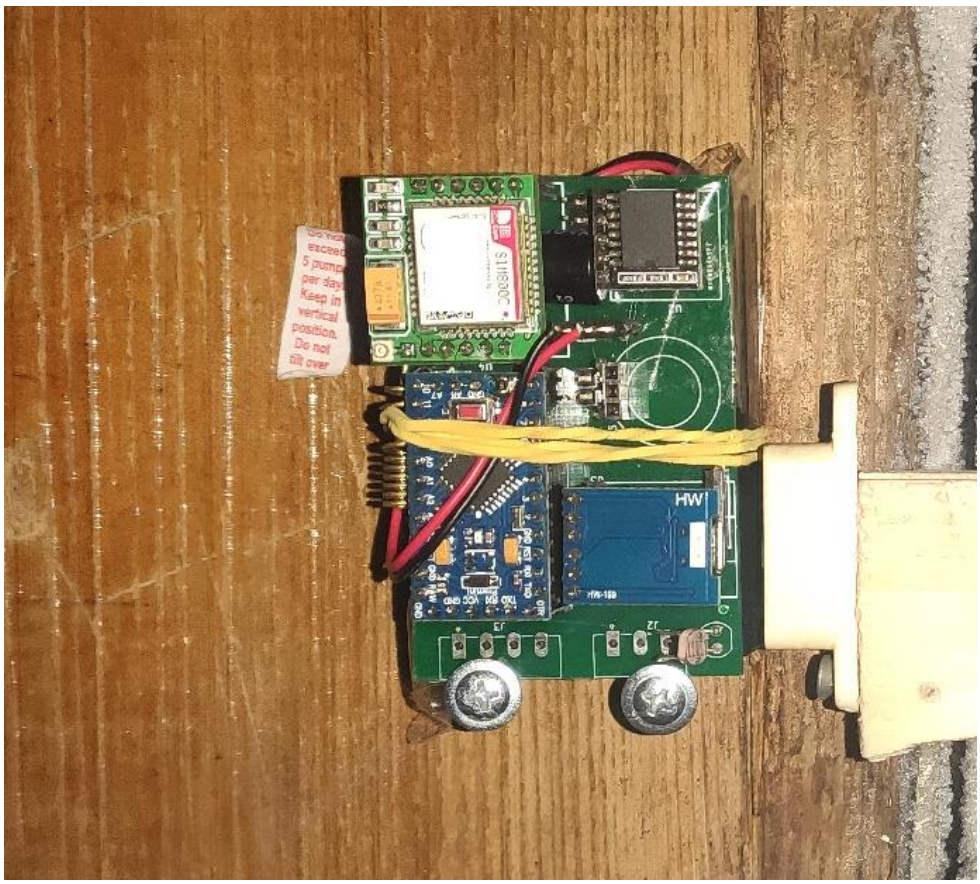

Рисунок 4.4 – Розміщення пристрою біля дверей

Висновки до розділу 4

У цьому розділі було докладно проаналізовано роботу застосунку GSMguard, який виступає важливим інструментом для швидкого та зручного налаштування охоронної системи перед її безпосереднім використанням. Завдяки широкому функціоналу додаток значно спрощує процес керування системою безпеки, дозволяючи користувачеві легко здійснювати моніторинг стану охоронюваного об’єкта. Інтерфейс програми розроблено таким чином, щоб навіть недосвідчені користувачі змогли інтуїтивно налаштувати всі необхідні параметри, отримувати сповіщення через SMS та контролювати роботу датчиків у реальному часі.

Розглянута система захисту під час тестування продемонструвала надійність у виявленні порушень та небажаних подій. Завдяки компактним розмірам охоронний модуль зручно монтується в малопомітних місцях – на дверях, вікнах або інших потенційно вразливих точках доступу до приміщення, що дозволяє забезпечити ефективний контроль без зайвого втручання в інтер’єр.

Суттєвою перевагою системи є можливість індивідуального налаштування режимів тривоги, параметрів повідомлень та інших функцій, що дозволяє адаптувати її під конкретні потреби користувача. Це забезпечує гнучкість і підвищує рівень безпеки завдяки своєчасному інформуванню про потенційні загрози.

Під час експлуатації були виявлені деякі обмеження, зокрема з боку датчиків, чутливість яких знижується при дії низьких температур. У зв’язку з цим у подальших версіях системи передбачається оновлення апаратної складової шляхом використання більш температуростійких компонентів.

Загалом проведене тестування та аналіз функціональних можливостей підтверджують, що система є ефективним інструментом для забезпечення безпеки, а її гнучка архітектура дозволяє адаптувати її під різні сценарії використання – від домашнього застосування до охорони гаражів, офісів чи складських приміщень.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи спроєктована та реалізована GSM-охоронна система на базі мікроконтролера Arduino.

Пристрій призначений для виявлення стороннього втручання на об’єкті та оперативного інформування користувача через SMS за допомогою GSM-зв’язку. Мета кваліфікаційної роботи створення простого, бюджетного, але функціонального рішення для базової охорони приміщення.

Щоб досягти цієї мети, спочатку були розглянуті та проаналізовані наявні на ринку охоронні системи, досліджено особливості роботи існуючих аналогів та виявлено їх переваги й недоліки. Далі були сформульовані технічні вимоги до розроблюваного пристрою та обрано необхідні компоненти, включаючи датчики руху, відкриття дверей і температури. На основі цих даних була створена схема системи, написано програмне забезпечення та проведено налаштування модулів.

Особливу увагу було приділено мобільному застосунку **GSMguard**, який суттєво спрощує процес налаштування системи. Завдяки інтуїтивному інтерфейсу користувач може швидко змінити параметри, подивитися графік температур, зчитати події, що відбувалися, або змінити налаштування сповіщення. Ця функція є дуже зручною, адже вона дозволяє уникнути перепрошивання пристрою кожного разу при зміні параметрів.

Тестування показало, що система добре справляється зі своїми завданнями. У середньому, протягом дня система надсилає 4 – 6 SMS залежно від активності біля об’єкта. Проблем із живленням не виникало – три батарейки типу AA забезпечують автономну роботу на декілька місяців. Єдине, що варто доопрацювати – це стабільність роботи датчиків при низьких температурах. Наприклад, датчик руху іноді спрацьовує хибно, а температурний сенсор показує некоректні значення при сильному морозі. Уникнути цих недоліків та покращити систему, можливо замінивши датчики на більш надійні моделі, стійкі до температурних перепадів. Також варто розглянути можливість використання більш потужного GSM-модуля або навіть використати інший спосіб передачі сповіщень для стабільнішого зв’язку в сільській місцевості.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Охоронна система [Електронний ресурс]:

https://uk.wikipedia.org/wiki/%D0%9E%D1%85%D0%BE%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0 (дата звернення 12.01.2025)

2. Охоронна сигналізація – які бувають та в чому різниця [Електронний ресурс]:

https://tech.gas.ua/blog/25?srsltid=AfmBOop16jM2wJCJ-oWR6XBeVDb9WZzehGGi1B5pbM8TlWvOs6yOBizC (дата звернення 12.01.2025)

3. Arduino [Електронний ресурс]:

https://uk.wikipedia.org/wiki/Arduino (дата звернення 12.01.2025)

4. Ajax Systems [Електронний ресурс]:

https://ajax.systems/ua/about/ (дата звернення 12.01.2025)

5. GSM-сигналізації Ajax [Електронний ресурс]:

https://ajax.systems/ua/product-categories/intrusion-protection/
(дата звернення 12.01.2025)

6. Модернізація та реновація [Електронний ресурс]:

https://ajax.systems/ua/upgrades-and-retrofits/ (дата звернення 12.01.2025)

7. MotionProtect Jeweller [Електронний ресурс]:

https://ajax.systems/ua/products/motionprotect/ (дата звернення 12.01.2025)

8. Ajax GlassProtect [Електронний ресурс]:

https://ajax.systems/ua/products/glassprotect/ (дата звернення 12.01.2025)

9. Магнітно-контактні датчики Ajax [Електронний ресурс]:

https://ajax.systems/ua/products/doorprotect/ (дата звернення 12.01.2025)

10. Пожежні датчики Ajax [Електронний ресурс]:

https://ajax.systems/ua/products/fireprotect-2-smoke-heat-co/

(дата звернення 12.01.2025)

11. Ajax LeaksProtect [Електронний ресурс]:
https://ajax.systems/ua/products/leaksprotect/ (дата звернення 12.01.2025)

12. Risco Group [Електронний ресурс]:

https://riscogroup.com/en/about/ (дата звернення 12.01.2025)

13. Agility 4 [Електронний ресурс]:

https://riscogroup.com/en/rscm_product/agility-4/ (дата звернення 12.01.2025)

14. Risco LightSYS 2 [Електронний ресурс]:

https://riscogroup.com/en/rscm_product/lightsys-2/ (дата звернення 12.01.2025)

15. Risco ProSYS Plus [Електронний ресурс]:

https://www.archiexpo.com/prod/risco-group-142571.html
(дата звернення 12.01.2025)

16. Risco RK150DTG3 [Електронний ресурс]:
https://riscogroup.com/en/rscm_product/lunar/ (дата звернення 12.01.2025)

17. Flood Detector [Електронний ресурс]:

https://riscogroup.com/en/rscm_product/flood-detector/ (дата звернення 12.01.2025)

18. Wireless LuNAR FireWatch Smoke [Електронний ресурс]:
https://riscogroup.com/en/rscm_product/wireless-lunar-firewatch/
(дата звернення 12.01.2025)

19. Glass Break Detectors [Електронний ресурс]:

https://riscogroup.com/en/rscm_product/glass-break-detectors/
(дата звернення 12.01.2025)

20. Arduino Pro Mini [Електронний ресурс]:

https://docs.arduino.cc/retired/boards/arduino-pro-mini/ (дата звернення 20.03.2025)

21. Сенсор руху MH-SR602 MINI [Електронний ресурс]: https://www.sinoning.com/hc-sr602-datasheet/?srsltid=AfmBOorsqWwwrBBtuH1vUoj7ATE7EVEZ79x4P4tcUfWf7TwyIC8IFYP7 (дата звернення 20.03.2025)

22. SIM800C(L) [Електронний ресурс]: https://www.alldatasheet.com/datasheet-pdf/pdf/1741387/SIMCOM/SIM800C.html (дата звернення 20.03.2025)

23. DS3231 RTC [Електронний ресурс]: https://www.diyengineers.com/2022/04/28/ds3231-datasheet/ (дата звернення 20.03.2025)

24. CP2102 USB–UART TTL [Електронний ресурс]: https://www.alldatasheet.com/view.jsp?Searchword=Cp2102&gad_source=1&gad_campaignid=170327939&gbraid=0AAAADcdDU8hcVBeaTRo8oGKFg5PJEXdK&gclid=CjwKCAjwl_XBBhAUEiwAWK2hznoNEGt1oTLDPhYU7TuKPHVt2dqWPz7REVAJKNZ5Vxw0FWWEO8wpVBoC3hoQAvD_BwE (дата звернення 20.03.2025)

25. Arduino IDE [Електронний ресурс]: https://www.arduino.cc/en/software/ (дата звернення 20.03.2025)

26. GSMguard [Електронний ресурс]: https://github.com/AlexIII/gsmGuard/releases/tag/v1.3 (дата звернення 25.04.2025)

ДОДАТКИ

Додаток А

Технічна специфікація мікроконтролера Arduino Pro Mini

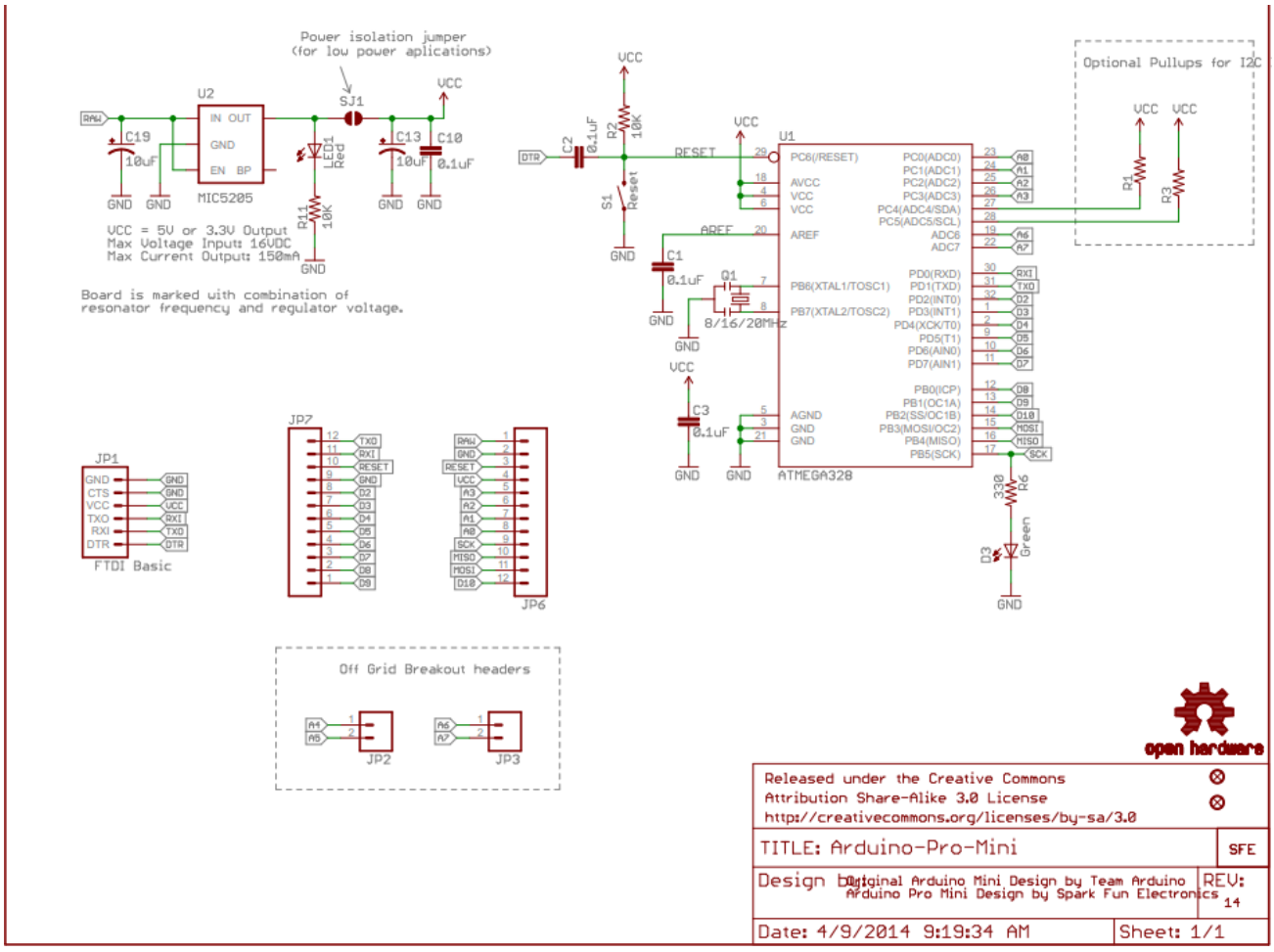

Рисунок А.1 – Схематичне зображення мікроконтролера Arduino Pro Mini

Таблиця А.1 – Технічні характеристики мікроконтролера Arduino Pro Mini

Microcontroller	ATmega328P *
Board Power Supply	3.35 -12 V (3.3V model) or 5 - 12 V (5V model)
Circuit Operating Voltage	3.3V or 5V (depending on model)
Digital I/O Pins	14
PWM Pins	6
UART	1
SPI	1
I2C	1
Analog Input Pins	6
External Interrupts	2
DC Current per I/O Pin	40 mA
Flash Memory	32KB of which 2 KB used by bootloader *
SRAM	2 KB *
EEPROM	1 KB *
Clock Speed	8 MHz (3.3V versions) or 16 MHz (5V versions)

Додаток Б

Креслення друкованої плати виробу

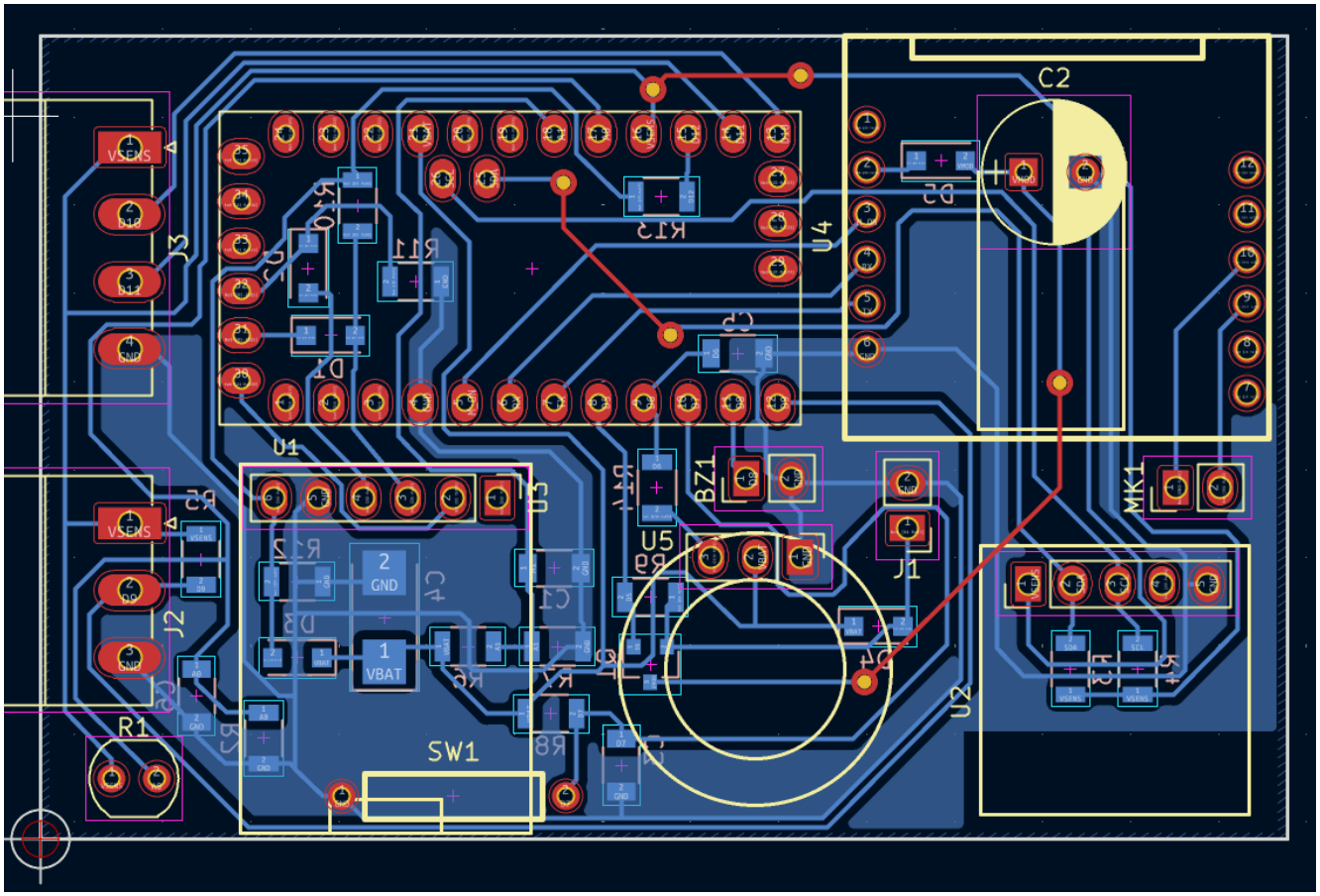

Рисунок Б.1 – креслення друкованої плати охоронної системи на базі мікроконтролера Arduino Pro Mini