

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерних систем та технологій

(повна назва кафедри)

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

**«Розробка побутової охоронної системи з підвищеною безпекою»**

(тема кваліфікаційної роботи українською мовою)

**«Development of a household security system with increased security»**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання

спеціальності 123 Комп'ютерна інженерія

(код, назва спеціальності)

Освітня програма Комп'ютерна інженерія

(назва)

Таранюк Олександр Андрійович

(прізвище, ім'я, по-батькові здобувача)

Керівник канд. ф.-м. наук, доц. Шугайло Ю.Б.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент канд. тех. наук, доц. Ларін Д.Г.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

комп'ютерних систем та технологій

№     від    .   .   .20    р.

Завідувач кафедри

Юрій ГУНЧЕНКО

(підпис)

(ім'я, прізвище)

Захищено на засіданні ЕК №    

протокол №     від    .   .   .20    р.

Оцінка     /     /    

(за національною шкалою / шкалою ECTS / бали)

Голова ЕК

Світлана АНТОЩУК

(підпис)

(ім'я, прізвище)

## АНОТАЦІЯ

У кваліфікаційній роботі запропоновано та реалізовано комплексну систему для виявлення несанкціонованого доступу до приміщень. Проведено огляд сучасних комп'ютеризованих рішень у сфері побутової охоронної сигналізації та обґрунтовано вибір архітектури з використанням програмно-апаратних модулів для віддаленого спостереження. Наведено функціональну блок-схему системи, розроблено електричну принципову схему блоку керування та детально описано алгоритм її роботи. Створено програмне забезпечення, яке гарантує стабільну взаємодію всіх компонентів сигналізації.

## **ANNOTATION**

The thesis proposes and implements a comprehensive system for detecting unauthorised access to premises. A review of modern computerised solutions in the field of domestic security alarms is carried out and the choice of architecture using software and hardware modules for remote monitoring is justified. The functional block diagram of the system is presented, the electrical schematic diagram of the control unit is developed, and the algorithm of its operation is described in detail. The software that guarantees stable interaction of all alarm components is created.

## ЗМІСТ

	Стор.
ПЕРЕЛІК СКОРОЧЕНЬ.....	6
ВСТУП .....	7
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	8
1.1 Вимоги до побутової охоронної системи .....	8
1.2 Аналіз потенційних варіантів реалізації поставленого завдання .....	9
1.3 Огляд існуючих комерційних охоронних рішень.....	12
1.3.1 Бездротова система охорони MAKS PRO .....	12
1.3.2 Комплект GSM-сигналізації Kit GSM .....	13
1.3.3 Стартовий комплект безпеки Ajax StarterKit .....	14
1.3.4 Внутрішня сигналізація Ajax HomeSiren.....	15
1.3.5 Результати порівняльного аналізу СОС .....	16
2 ПРОЄКТНА ЧАСТИНА .....	17
2.1 Розробка структури системи побутової охоронної сигналізації.....	17
2.2 Аргументація вибору компонентної бази системи побутової охоронної сигналізації .....	20
2.2.1 Керуюча плата на базі мікроконтролера .....	20
2.2.2 Модуль датчика руху.....	22
2.2.3 Датчик відкриття дверей.....	23
2.2.4 Сенсор контролю розбиття скла.....	25
2.2.5 Комунікаційний модуль GSM SIM800L.....	26
2.2.6 Блок контролю заряду/розряду АКБ.....	28
2.2.7 Модуль звукового сповіщення (п'єзодинаміка) .....	29
2.2.8 Інформаційний дисплей .....	30
2.2.9 Інтерфейсний модуль I <sup>2</sup> C .....	31
2.2.10 Модуль клавіатури.....	32
2.3 Проєктування електричної принципової схеми системи побутової охоронної сигналізації.....	33

2.3.1 Обґрунтування вибору середовища проектування електричних схем .....	33
2.3.2 Розробка принципової схеми пристрою .....	34
3 ПРАКТИЧНА ЧАСТИНА .....	37
3.1 Розробка алгоритмічної моделі функціонування системи побутової охоронної сигналізації.....	37
3.2 Налаштування середовища для розробки ПЗ.....	41
3.2.1 Середовище розробки програмного коду для мікроконтролера.....	41
3.2.2 Підключення бібліотеки для взаємодії з GSM-модулем .....	42
3.3 Реалізація ПЗ системи побутової охоронної сигналізації.....	43
3.3.1 Код для опитування клавіатури.....	43
3.3.2 Код для виведення інформації на LCD-дисплей .....	45
3.3.3 Код для обміну даними з GSM-модулем.....	46
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А.....	52
ДОДАТОК Б .....	62

## ПЕРЕЛІК СКОРОЧЕНЬ

АЦП – аналого-цифровий перетворювач

БЖ – блок живлення

БК – блок керування

КС – комп'ютерна система

МК – мікроконтролер

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

СОС – система охоронної сигналізації

ШІМ – широтно-імпульсна модуляція

## ВСТУП

Щороку правоохоронні органи стикаються з численними випадками незаконного проникнення до житлових приміщень та вчиненням крадіжок, що становить гостру соціальну й економічну проблему. За офіційними даними, у 2024 році в Україні зафіксовано понад 1900 квартирних крадіжок [1].

Визнаним засобом зниження рівня злочинності в побутовій сфері є встановлення систем охоронної сигналізації. Завдяки своєму функціоналу такі системи не лише сповіщають власників і сусідів про спробу несанкціонованого доступу, а й можуть відлякати зловмисників звуковим оповіщенням. Особливо ефективними вважаються рішення з підключенням до централізованих пультів охоронних підприємств, що дозволяє миттєво інформувати відповідні служби.

Однак наявні на ринку пропозиції часто відрізняються високою вартістю, що обмежує їх доступність для широкого кола користувачів. Тому розробка економічно вигідної та надійної побутової охоронної системи, виконаної на основі сучасних мікропроцесорних технологій і компонентної бази, залишається актуальним завданням.

Метою даної кваліфікаційної роботи є створення комп'ютеризованої охоронної системи для цілодобового захисту житлових приміщень із використанням доступних апаратних рішень.

Для досягнення поставленої мети необхідно виконати такі завдання:

- 1) провести огляд і порівняльний аналіз існуючих на ринку аналогів;
- 2) розробити узагальнену структурну схему побутової охоронної системи;
- 3) скласти принципову електричну схему керуючого модуля;
- 4) описати алгоритм роботи ключових програмних компонентів;
- 5) розробити ПЗ, що забезпечить належну інтеграцію та функціонування всіх елементів системи.

## 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

У цьому розділі виконано детальне дослідження вимог до побутової охоронної системи та проведено огляд існуючих рішень на ринку з визначенням їх сильних і слабких сторін. Також розглянуто можливі підходи до реалізації поставленого завдання.

### 1.1 Вимоги до побутової охоронної системи

Система охоронної сигналізації (СОС) – це сукупність апаратних та програмних засобів, призначених для захисту приміщення від несанкціонованого проникнення всередину й зовні. Сучасні комплексні рішення об'єднують відеоспостереження, охоронну та пожежну сигналізацію, систему контролю доступу й інші спеціалізовані модулі, а також забезпечують їхню взаємодію.

Згідно з технічним завданням, розроблена СОС повинна:

- 1) працювати у режимі реального часу;
- 2) автоматично вмикати звуковий оповіщувач і розсилати повідомлення при виявленні спроби проникнення;
- 3) інформувати власника про всі спрацьовування датчиків;
- 4) забезпечувати віддалений моніторинг стану приміщення через бездротовий зв'язок;
- 5) відображати результати опитування датчиків на інтегрованому дисплеї.

Для коректного виконання цих функцій передбачається застосування датчиків руху, відкриття дверей та розбиття скла, а також вбудованого генератора звукових сигналів. Центральним елементом системи стане мікроконтролер, що здійснює опитування всіх датчиків та координує роботу сигналізації. Конкретний тип мікроконтролера обирається на основі порівняльної оцінки сучасних компонентів електронної бази.

Оскільки одним із ключових завдань є віддалене сповіщення користувача, необхідно визначити та інтегрувати відповідну технологію передачі даних, що забезпечить надійну передачу повідомлень у режимі онлайн.

## **1.2 Аналіз потенційних варіантів реалізації поставленого завдання**

Охоронна сигналізація являє собою об'єднаний комплекс технічних засобів, призначений для виявлення та нейтралізації загроз безпеці охоронюваних об'єктів. Для збору інформації про стан приміщення використовуються різноманітні датчики, які передають свої показники на центральну панель керування через відповідні канали зв'язку. Датчик – це пристрій, який встановлюється безпосередньо в контрольованій зоні та перетворює виявлені величини у формат, придатний для передачі та обробки. Оскільки більшість датчиків СОС фіксує неелектричні параметри, до їхньої роботи пред'являються підвищені вимоги щодо надійності та точності реагування [2].

Найчастіше в системах побутової сигналізації застосовуються такі типи датчиків [3, 4]:

- 1) інфрачервоні прилади для виявлення руху та присутності;
- 2) сенсори розбиття скла;
- 3) мікрохвильові детектори;
- 4) вібраційні сенсори;
- 5) фотоприймачі (фотоеlementи);
- 6) герконові (магнітні) датчики відкриття дверей і вікон;
- 7) ультразвукові детектори.

Основними функціями СОС є своєчасне виявлення спроб несанкціонованого проникнення, передача тривожного повідомлення на центральний пункт спостереження та активація виконавчих пристроїв

(освітлення, сирена тощо). Комплект системи зазвичай включає керуючий модуль, чутливі елементи, виконавчі пристрої та комунікаційні модулі.

Блок керування (БК) виступає центральним елементом СОС. Він побудований на базі мікроконтролера, прошивка якого реалізує весь набір необхідних алгоритмів обробки сигналів і керування пристроями. Через БК можна під'єднати зовнішній комп'ютер для реєстрації тривожних подій, автоматизованого збирання статистики та аналізу стану датчиків і загальної ефективності роботи сигналізації. За спрацьовування датчиків БК активує виконавчі прилади – вмикає освітлення, запускає звукову сирену або передає текстові/голосові повідомлення власнику об'єкта.

За способом передачі інформації СОС поділяються на три категорії:

- 1) дротові;
- 2) бездротові;
- 3) гібридні.

Раніше перевага віддавалася дротовим СОС через обмежену надійність радіоканалів та ризик втрати зв'язку, але сучасні бездротові технології вже забезпечують високий рівень безпеки, стабільності та простоту монтажу [2].

Дротові рішення застосовуються переважно там, де до датчиків необхідно подати живлення від зовнішнього джерела; їх будують за топологією «зірка», коли всі лінії з'єднані з центральним БК. Нові ж бездротові комплекси дозволяють оперативно розгорнути захист об'єкта без значних витрат на прокладання кабелів і проводять оновлення та обслуговування мінімальною участю техніків [5].

СОС складається з сукупності апаратних і програмних компонентів, що взаємодіють між собою для забезпечення захисту об'єкта від несанкціонованих втручань. У рамках кваліфікаційної роботи розглядається побутова охоронна сигналізація – мережа електронних приладів, об'єднаних єдиним центральним модулем керування, яка призначена для відбиття спроб проникнення злоумисників.

Загальносвітова тенденція до зростання кількості крадіжок та грабежів зумовлює підвищений попит на подібні рішення. Більшість злочинці віддають перевагу об'єктам, де відсутня охоронна система, проникаючи переважно через двері чи вікна. Типову архітектуру сигналізації наведено на рисунку 1.1, де показано основні вузли та їх взаємозв'язки [6].

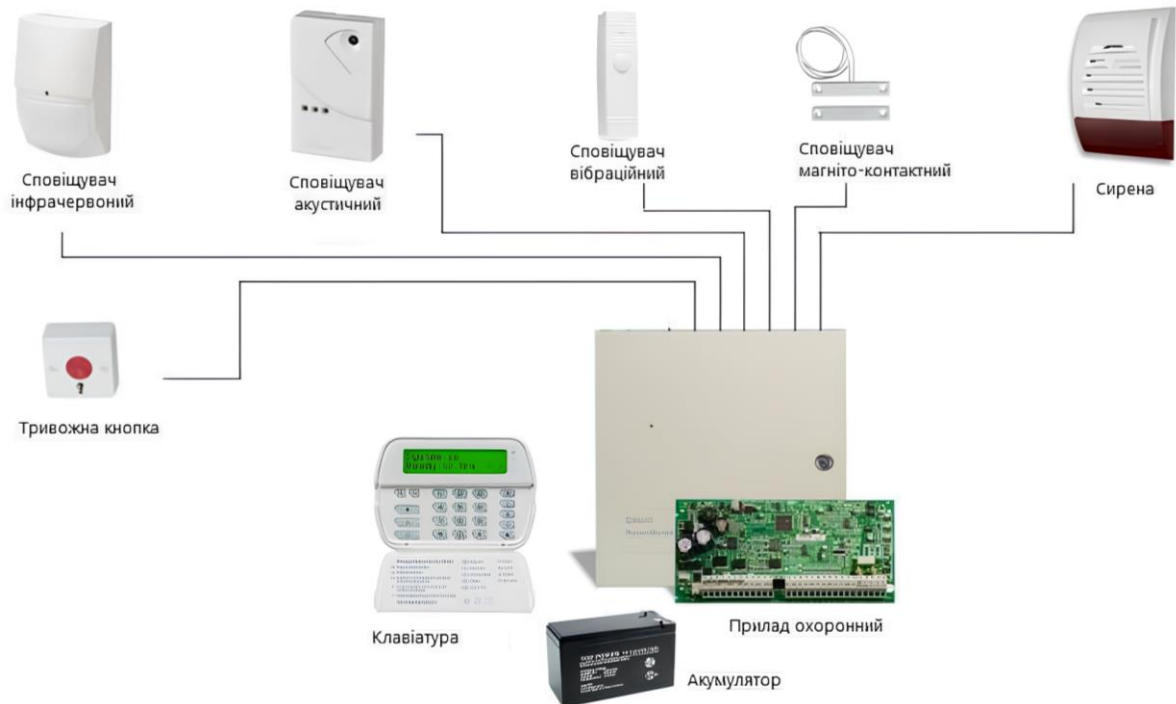


Рисунок 1.1 – Структура типової СОС

Статистичні дослідження підтверджують: просте встановлення сигналізації здебільшого відбиває наміри проникнути в приміщення. Навіть базові конфігурації системи суттєво знижують ймовірність вчинення квартирних крадіжок.

У разі спрацювання датчика СОС здатна миттєво активувати гучний звуковий сигнал і, за наявності відповідного модуля, надіслати тривожне повідомлення власнику. Сучасні технології передачі даних по радіоканалу забезпечують більшу стійкість до перешкод і розширюють зону покриття, що підвищує надійність системи охоронного спостереження [7].

### 1.3 Огляд існуючих комерційних охоронних рішень

Наразі на українському ринку представлено чимало побутових охоронних систем. Нижче наведено стислий огляд рішень від провідних виробників із описом їхніх ключових особливостей та технічних характеристик.

#### 1.3.1 Бездротова система охорони MAKS PRO

MAKS PRO – це рішення для охорони об’єктів середнього та великого масштабу, яке може працювати автономно або в зв’язці з централізованим пультом спостереження (рис. 1.2). Налаштування та керування здійснюються через мобільні додатки на платформах iOS та Android [8].

Однією з ключових особливостей MAKS PRO є можливість автоматичного переходу в енергоощадний режим при живленні від акумулятора у разі відключення основного джерела. Система підтримує мультимедійну передачу тривожних повідомлень через мережі GSM/GPRS.



Рисунок 1.2 – Система бездротової сигналізації MAKS PRO

MAKS PRO легко масштабується: до неї можна підключити до 200 пристроїв, включаючи датчики руху та розбиття скла, брелоки, клавіатури, сирени та інші елементи охоронного комплексу.

### 1.3.2 Комплект GSM-сигналізації Kit GSM

Kit GSM є бездротовим рішенням на базі GSM-технології, яке при виявленні несанкціонованого доступу автоматично формує тривожний сигнал, здійснює голосові виклики та відправляє SMS на задані телефонні номери (рис. 1.3).

Випускає цю модель підприємство ATIS. Керувати сигналізацією можна двома способами: за допомогою радіокерованого брелока або через мобільний додаток на смартфоні [9].



Рисунок 1.3 – Система бездротової сигналізації Kit GSM

Основним каналом обміну даними в Kit GSM виступає GSM-мережа. Для забезпечення безперервної роботи навіть за відключення живлення передбачено вбудований літійовий акумулятор. Інформаційний інтерфейс реалізовано через рідкокристалічний дисплей, на якому відображаються ключові параметри та статус системи.

### 1.3.3 Стартовий комплект безпеки Ajax StarterKit

StarterKit – це охоронна система, призначена для захисту житлових або комерційних приміщень від несанкціонованого проникнення через двері та вікна (рис. 1.4). Сигнали тривоги передаються на центральний пульт моніторингу та безпосередньо на смартфон користувача за допомогою Ethernet і GSM-зв'язку. Управління системою здійснюється через мобільний застосунок [10].



Рисунок 1.4 – Система безпеки StarterKit від компанії Ajax

Принцип роботи полягає в постійному моніторингу стану встановлених датчиків, які фіксують відкриття дверей, вікон та рух у приміщенні. Головний модуль (хаб) StarterKit відповідає за централізовану обробку подій і передачу тривожних сповіщень.

Набір базових компонентів можна розширити додаванням сенсорів для виявлення затоплення або задимлення. Також до системи можна інтегрувати відеоспостереження та елементи «розумного дому» для розширення функціональних можливостей.

### 1.3.4 Внутрішня сигналізація Ajax HomeSiren

HomeSiren – це бездротовий внутрішній модуль для побутової сигналізації, призначений для генерації звукового попередження при спрацьовуванні датчиків (рис. 1.5). Прилад встановлюється всередині приміщення, щоб відлякати порушників і своєчасно інформувати про загрозу.



Рисунок 1.5 – Бездротова кімнатна охоронна сигналізація HomeSiren

Система дозволяє налаштовувати гучність і тривалість сигналу відповідно до потреб користувача. За бажанням до корпусу сирени можна підключити зовнішній світлодіодний індикатор, що розміщується за межами контрольованої зони та візуально відображає стан охоронної системи [11].

### **1.3.5 Результати порівняльного аналізу СОС**

Підсумовуючи проведений огляд, варто зазначити, що практично всі розглянуті рішення при спрацьовуванні датчиків формують звуковий або світловий сигнал. Деякі з високобюджетних моделей додатково підтримують відправлення повідомлень безпосередньо на смартфон власника.

Таким чином, результати аналізу свідчать, що одним із найбільш перспективних напрямків розвитку побутової охоронної сигналізації є застосування GSM-зв'язку, який забезпечує оперативне віддалене інформування про надзвичайні події.

## 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Розробка структури системи побутової охоронної сигналізації

Під час розробки системи охоронної сигналізації застосовано модульний підхід, який передбачає розділення загальної архітектури на окремі функціональні частини, кожна з яких виконує визначене завдання. Загальну структурно-функціональну модель побутової системи охоронної сигналізації представлено на рисунку 2.1.

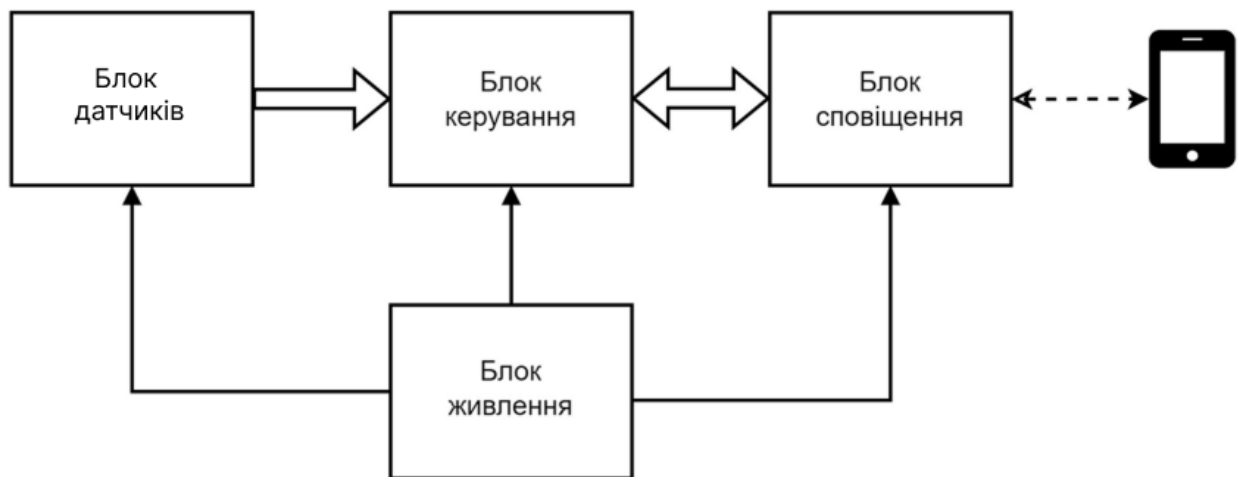


Рисунок 2.1 – Структурно-функціональна модель побутової системи охоронної сигналізації

Проектована система передбачає наявність таких основних функціональних блоків:

- 1) блок датчиків;
- 2) блок керування;
- 3) блок сповіщення;
- 4) блок живлення.

Оскільки головною функцією охоронної сигналізації є моніторинг ситуації в приміщенні, першим обов'язковим елементом є набір сенсорів,

згрупованих у відповідний модуль. Для обробки сигналів, отриманих з сенсорного блоку, необхідна керуюча частина, що реалізується у вигляді контролера. Забезпечення роботи системи в автономному режимі потребує включення джерела енергії (акумулятора) із системою стабілізації та зарядки, які виділяються в окремий БК. Для забезпечення зворотного зв'язку та сигналізації про порушення призначено окремий блок сповіщення [12]. Деталізовану структурну схему охоронної системи можна побачити на рисунку 2.2.

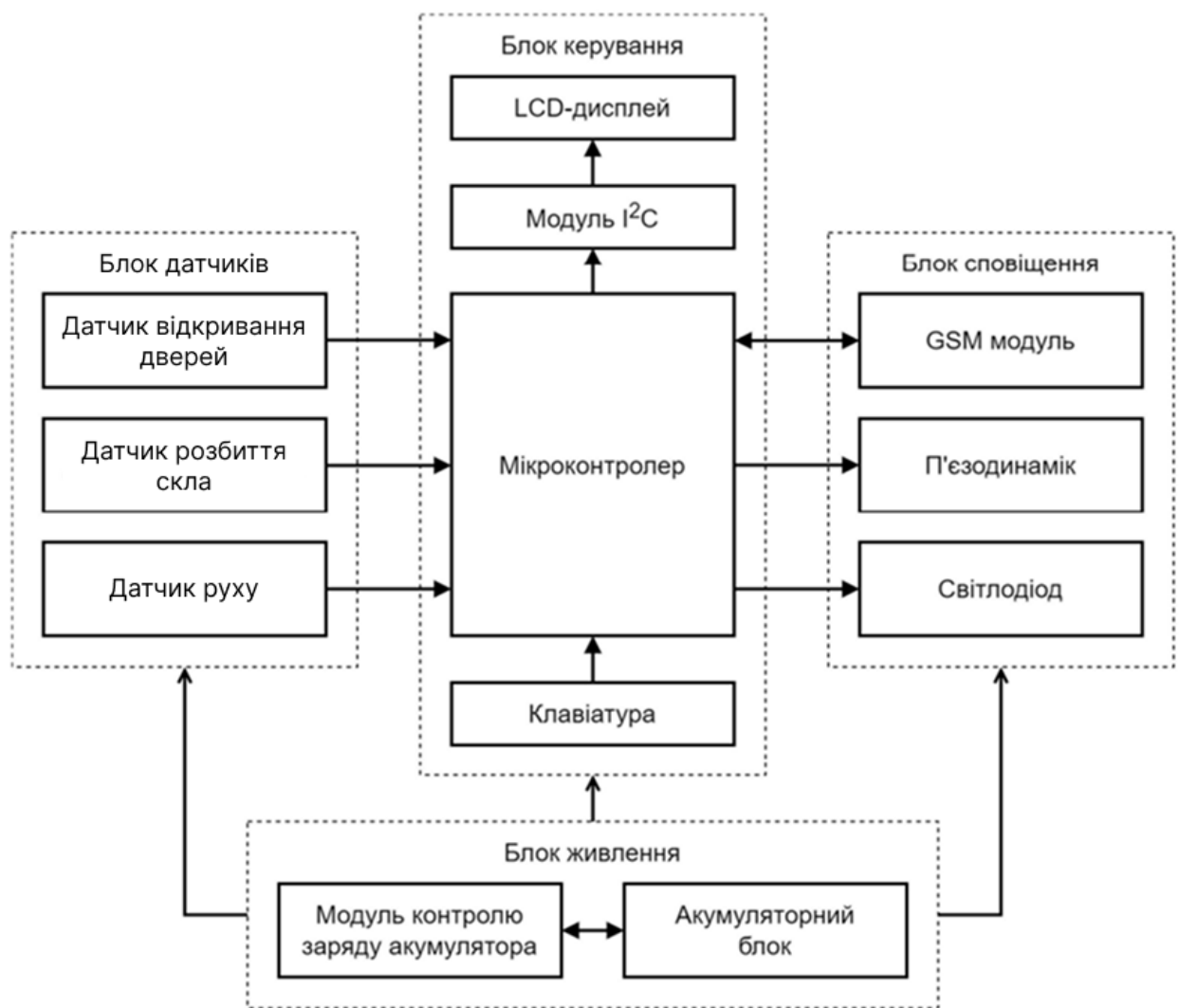


Рисунок 2.2 – Деталізована структурна схема системи побутової охоронної сигналізації

Блок датчиків виконує роль джерела інформації про поточний стан об'єкта, що охороняється, передаючи зібрані дані до керуючого модуля. До його складу входять такі типи датчиків:

- 1) датчик відкриття дверей;
- 2) датчик розбиття скла;
- 3) інфрачервоний сенсор руху.

Цей набір дозволяє зібрати повну картину подій у зоні спостереження та своєчасно виявити несанкціоноване проникнення. Живлення сенсорів може здійснюватися автономно, що дає змогу гнучко розміщувати їх у приміщенні, не залежачи від центрального джерела живлення. Наприклад, сенсори на дверях часто живляться від гальванічних елементів або можуть функціонувати за рахунок вбудованої логіки без зовнішнього живлення.

Інфрачервоний сенсор забезпечує виявлення руху в кімнаті, реагуючи на теплове випромінювання тіла. При фіксації рухомого об'єкта він миттєво надсилає сигнал до керуючого блоку.

Датчик відкриття дверей реєструє зміну положення дверей, що є критично важливою ознакою проникнення. Датчик скла орієнтований на виявлення розбиття вікна, аналізуючи характерні звукові хвилі, притаманні цьому типу події.

Таким чином, обраний перелік сенсорів дозволяє надійно охопити всі потенційні шляхи проникнення та вчасно фіксувати рух у межах приміщення.

Керуючий модуль об'єднує в собі наступні компоненти:

- 1) мікроконтролерна плата;
- 2) матрична клавіатура;
- 3) модуль I<sup>2</sup>C;
- 4) рідкокристалічний дисплей.

Центральним елементом системи є мікроконтролер, що забезпечує координацію роботи всіх частин сигналізації. За допомогою клавіатури та дисплея реалізовано базовий інтерфейс користувача – для введення коду, постановки на охорону та зняття з неї, а також для зміни параметрів,

наприклад, пароля. Клавіатура має 12 кнопок, організованих у вигляді сітки 3x4. Застосування I<sup>2</sup>C дозволяє зменшити кількість з'єднань для підключення дисплея.

Блок сповіщення включає:

- 1) модуль GSM-зв'язку;
- 2) звуковий сигналізатор (п'єзодинамік);
- 3) світлодіодний індикатор.

GSM-модуль забезпечує відправлення повідомлень до мобільного пристрою власника або диспетчерського пункту. П'єзодинамік виконує функцію гучного оповіщення про загрозу, а світлодіод слугує для візуальної індикації стану тривоги.

## **2.2 Аргументація вибору компонентної бази системи побутової охоронної сигналізації**

### **2.2.1 Керуюча плата на базі мікроконтролера**

В якості основного модуля керування у розробленій системі охоронної сигналізації обрано мікроконтролерну платформу Arduino UNO (рис. 2.3). Такий вибір обґрунтовано доступною ціною плати, а також великою кількістю сумісних модулів і відкритих бібліотек, що значно спрощує розробку і впровадження нових функцій [13].

Arduino UNO забезпечує зв'язок із зовнішніми компонентами через чотирнадцять цифрових портів загального призначення (шість з яких можуть працювати в режимі ШІМ), а також шість аналогових входів. Крім того, вона має USB-роз'єм для програмування та порт I<sup>2</sup>C. Живлення може подаватися як через USB, так і через роз'єм для зовнішнього джерела. З огляду на вимогу автономності системи, передбачену технічним завданням, живлення буде організовано через акумуляторну батарею.

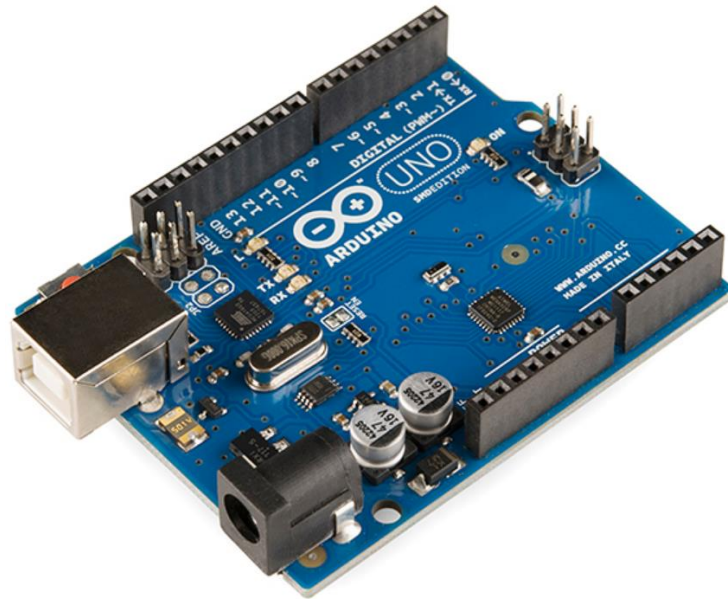


Рисунок 2.3 – Зовнішній вигляд плати Arduino UNO

Плата функціонує на базі 8-бітного мікроконтролера ATmega328P, розташування та функції виводів якого наведені на рисунку 2.4.

(PCINT14/ $\overline{\text{RESET}}$ ) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 ( $\overline{\text{SS}}$ /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Рисунок 2.4 – Призначення контактів мікроконтролера ATmega328P

Мікроконтролер ATmega328P побудований на розширеній архітектурі RISC, що дозволяє досягати хороших показників продуктивності при

компактному розміщенні коду. Він оснащений 32 КБ флеш-пам'яті, з яких 0,5 КБ відведено під завантажувач. Крім того, пристрій має 2 КБ оперативної пам'яті SRAM та 1 КБ енергонезалежної пам'яті EEPROM [14].

Тактова частота роботи становить 16 МГц, що забезпечується кварцовим резонатором, вбудованим у плату. Одним із ключових чинників вибору саме цього мікроконтролера стала наявність вбудованого 10-бітного аналого-цифрового перетворювача (АЦП), необхідного для обробки сигналів із датчиків. До додаткових можливостей контролера відносяться підтримка таких інтерфейсів, як I<sup>2</sup>C, SPI та UART, що забезпечують гнучкість у підключенні периферійних пристроїв.

### 2.2.2 Модуль датчика руху

Для реалізації виявлення стороннього руху в межах контрольованої зони в проєктованій охоронній системі обрано інфрачервоний піроелектричний сенсор руху PIR на основі модуля HC-SR501 (рис. 2.5) [15]. Цей сенсор здатен визначати присутність людей у межах свого поля зору, фіксуючи зміну інфрачервоного випромінювання, що надходить від теплокровних об'єктів. Також він може реагувати на інші теплові джерела.



Рисунок 2.5 – PIR-модуль HC-SR501

Модуль обладнаний лінзою Френеля, що концентрує інфрачервоне випромінювання на піроелектричному елементі. Даний сенсор є пасивним, оскільки сам не генерує випромінювання, а лише сприймає енергію, яку випромінюють об'єкти, що знаходяться в полі його дії. Чутлива частина пристрою складається з двох окремих приймачів. Вбудована мікросхема аналізує сигнали, які надходять з обох частин, і при виявленні асиметрії у випромінюванні фіксує наявність руху. Основні технічні параметри модуля датчика руху наведені у таблиці 2.1.

Таблиця 2.1 – Технічні параметри датчика руху

Параметр	Значення
Струм	50 мкА
Напруга	4,5–20 В
Вихідна напруга	3,3 В

Модуль HC-SR501 є компактним, простим в інтеграції, має мале енергоспоживання і високу стабільність роботи, що робить його зручним для застосування в автономних охоронних пристроях. При цьому слід враховувати, що ефективність сенсора знижується з віддаленням об'єкта. Також небажано розміщувати його поблизу джерел тепла або інтенсивного світла, що можуть викликати помилкові спрацювання через вплив на чутливу поверхню лінзи.

### 2.2.3 Датчик відкриття дверей

Серед ключових сенсорів, які застосовуються в конструкції системи охоронної сигналізації, особливе значення має датчик відкриття дверей. У запропонованій реалізації використовується герконовий датчик типу МС-38,

що функціонує на основі впливу магнітного поля [16]. Зовнішній вигляд цього пристрою наведено на рисунку 2.6.



Рисунок 2.6 – Зовнішній вигляд герконового датчика відкриття дверей МС-38

Даний датчик має потужність до 10 Вт та здатен працювати при максимальному струмі до 500 мА. Його спрацьовування відбувається при розриві відстані між герконом та магнітом на приблизно 18 мм, з допустимим відхиленням у межах  $\pm 6$  мм.

Однією з особливостей цього сенсора є те, що він належить до типу нормально замкнених пристроїв, тобто електричне з'єднання між контактами зберігається доти, доки магніт розміщений поруч із герконом. У момент, коли двері відкриваються і магніт віддаляється, контакт розмикається, що фіксується системою як сигнал тривоги або подія доступу.

## 2.2.4 Сенсор контролю розбиття скла

Для реалізації функції виявлення розбиття скла в рамках розробленої охоронної системи використано акустичний модуль KY-037 [17]. Цей пристрій може бути налаштований на реагування лише на звуки певної інтенсивності. Вигляд модуля подано на рисунку 2.7.

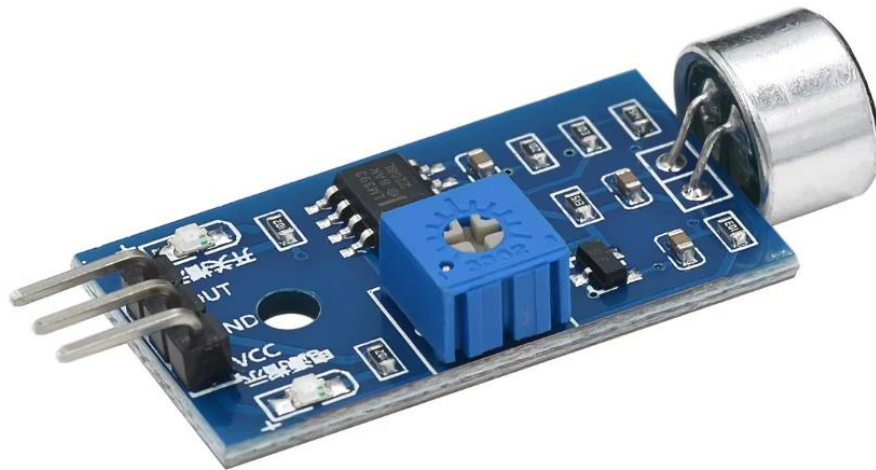


Рисунок 2.7 – Зовнішній вигляд звукового датчика KY-037

Робота датчика KY-037 ґрунтується на реєстрації механічних коливань мембрани вбудованого мікрофона, які виникають під впливом звукових хвиль. Зміни ємності конденсатора, пов'язані з цими коливаннями, призводять до варіації напруги на його виходах. Ця зміна сигналу дозволяє сенсору фіксувати звукові події у навколишньому середовищі.

Модуль ідеально підходить для розпізнавання різких та гучних звуків – таких, як звук битого скла, що робить його ефективним у виявленні спроб несанкціонованого проникнення. При цьому малопотужні сигнали, зокрема фонові розмови або слабе шумове тло, залишаються поза його чутливістю, що зменшує ймовірність помилкових спрацьовувань.

### 2.2.5 Комунікаційний модуль GSM SIM800L

Для реалізації функції повідомлення власника про тривожну подію в зоні, яка перебуває під охороною, обрано GSM-модуль SIM800L, вигляд якого наведено на рисунку 2.8. Даний пристрій виконує ті самі завдання, що і звичайний мобільний телефон, завдяки чому є придатним для інтеграції в охоронну систему.



Рисунок 2.8 – Зовнішній вигляд модуля SIM800L

Модуль SIM800L здатен виконувати функції дзвінків, надсилання текстових повідомлень, а також забезпечує передачу даних через мобільну мережу з підтримкою таких протоколів, як TCP/IP і GPRS. Пристрій функціонує у чотирьох діапазонах GSM-зв'язку, що дозволяє йому працювати в умовах різних мобільних мереж. В основі модуля лежить мікросхема SIM800L від виробника SimCom [18].

Стандартна конфігурація пристрою включає вбудовану антену, проте за потреби можливе під'єднання зовнішньої, що покращує стабільність і

потужність сигналу. SIM-картка встановлюється у спеціальний слот, розміщений на зворотній стороні плати. Для підключення до контролера застосовується послідовний інтерфейс UART, який забезпечує швидкість обміну даними до 11520 біт/с.

Оскільки логічний рівень сигналів SIM800L становить 2,1–3,1 В, а мікроконтролери сімейства ATmega зазвичай працюють з рівнем 5 В, пряме підключення лінії передачі може пошкодити модуль. З цієї причини рекомендується використовувати дільник напруги на базі двох резисторів номіналами 10 кОм і 10 кОм для узгодження рівнів сигналу на вході RXD.

Перед використанням модуль необхідно підготувати – вставити SIM-картку у відповідний слот та під'єднати антену. Призначення контактів модуля SIM800L показано на рисунку 2.9.



Рисунок 2.9 – Призначення контактів GSM-модуля SIM800L

Ще однією важливою особливістю є високий пусковий струм споживання, який може досягати 2 А, через що живлення від контролера є неприйнятним. Рекомендоване джерело живлення – літєвий акумулятор або стабілізоване джерело з напругою в межах 3,3–4,4 В.

## 2.2.6 Блок контролю заряду/розряду АКБ

Відповідно до технічного завдання, охоронна сигналізація забезпечує автономну роботу від акумулятора. Для його регулярного заряджання використовується зарядний модуль на основі мікросхеми TP4056 (рис. 2.10).

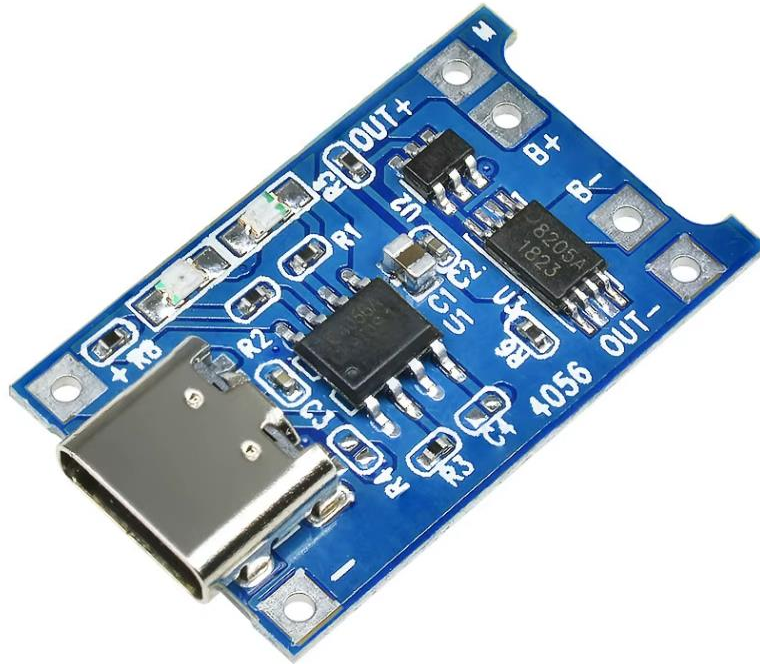


Рисунок 2.10 – Модуль зарядного пристрою TP4056

Цей модуль призначений для заряджання літійових акумуляторів, а сам процес має аналогію із заряджанням мобільного телефону. У процесі заряджання світлодіоди на платі сигналізують про стан батареї: під час заряджання світиться один індикатор, а після досягнення повного заряду – інший [19].

Електроживлення до модуля може подаватися через порт USB Type-C, або ж альтернативно – через відповідні контактні площадки, до яких при необхідності можна під'єднати проводи пайкою. Така гнучкість у підключенні робить TP4056 зручним рішенням для вбудованих систем, що працюють автономно.

### 2.2.7 Модуль звукового сповіщення (п'єзодинаміка)

У складі розроблюваної охоронної системи для звукової індикації у разі спрацювання сенсорів вирішено застосувати п'єзодинамічний модуль. Його завданням є перетворення сигналу від мікроконтролера на звукове повідомлення, яке сповіщає про тривожну подію. На рисунку 2.11 представлено зовнішній вигляд цього модуля.



Рисунок 2.11 – Зовнішній вигляд модуля п'єзодинаміка

Конструктивно п'єзодинамічний елемент складається з металевої пластини, поверхня якої покрита спеціальним керамічним шаром, що має провідні властивості. Робота пристрою ґрунтується на п'єзоелектричному ефекті: під дією електричного струму відбувається деформація матеріалу, яка викликає коливання, що в свою чергу генерують звукові хвилі [20].

Для формування звуку з певною частотою модуль обладнаний вбудованим генератором, який дозволяє створювати сигнал з необхідними параметрами. У таблиці 2.2 подано основні технічні характеристики даного модуля.

Таблиця 2.2 – Технічні параметри модуля п'єзодинаміка

Параметр	Значення
Частота звуку	2300 Гц
Струм	до 30 мА
Напруга	5 В

### 2.2.8 Інформаційний дисплей

В рамках розробленої охоронної системи рідкокристалічний дисплей (LCD) виконує роль засобу виведення інформації, зокрема інформує про стан підключених сенсорів, а також забезпечує взаємодію з користувачем під час введення пароля при активації або деактивації охорони. На рисунку 2.12 зображено зовнішній вигляд використаного дисплейного модуля.

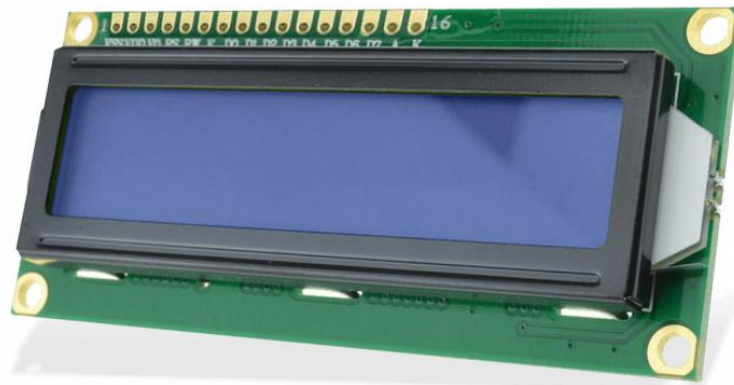


Рисунок 2.12 – Зовнішній вигляд LCD-дисплея

Керування дисплеєм здійснюється за допомогою контролера HD44780, який забезпечує обробку команд, що надходять з мікроконтролера Arduino. Обмін інформацією реалізується через інтерфейс I<sup>2</sup>C, що дозволяє зменшити кількість необхідних з'єднань між дисплеєм та основною платою управління, що є важливим для компактності схеми [21].

## 2.2.9 Інтерфейсний модуль I<sup>2</sup>C

Для оптимізації кількості задіяних виводів на мікроконтролері та спрощення взаємодії з рідкокристалічним дисплеєм у розробленій системі вирішено використати інтерфейсний модуль I<sup>2</sup>C, створений на основі мікросхеми PCF8574T, яка виконує функцію контролера вводу/виводу. На рисунку 2.13 показано зовнішній вигляд цього модуля.

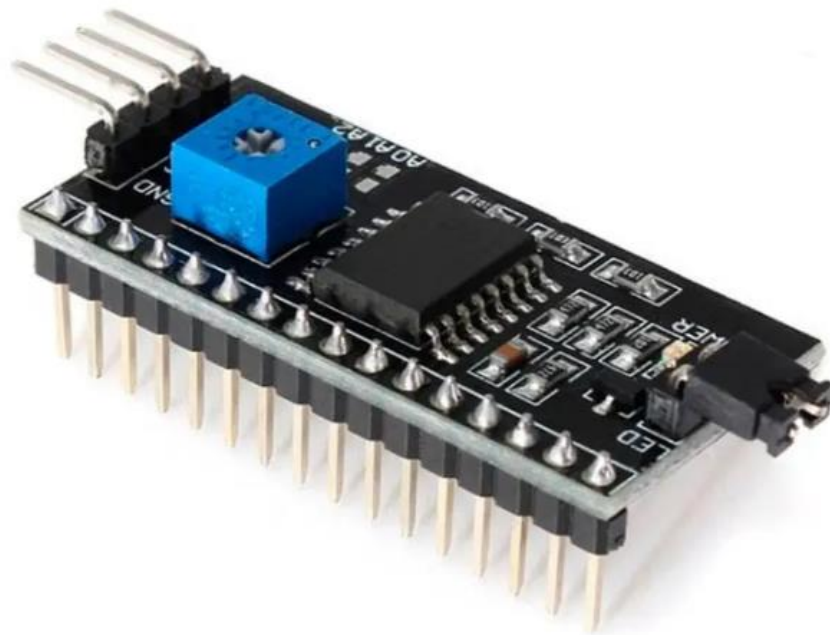


Рисунок 2.13 – Зовнішній вигляд інтерфейсного модуля I<sup>2</sup>C

Передача даних до дисплея здійснюється через лише дві сигнальні лінії: SDA (дані) та SCL (синхронізація), що дозволяє значно скоротити кількість підключень у порівнянні зі стандартним паралельним з'єднанням [22]. Лінія SDA під'єднується до відповідного входу обміну даними на платі мікроконтролера, а SCL – до одного з цифрових портів Arduino. Саме через ці контакти дані надходять до модуля I<sup>2</sup>C, а потім передаються на дисплей для відображення.

### 2.2.10 Модуль клавіатури

У проєктованій системі управління доступом клавіатура використовується як засіб введення секретного коду для деактивації охоронного режиму. Також за її допомогою можливо виконувати інші функції, зокрема оновлення пароля або зміна параметрів конфігурації системи. Для реалізації цих функцій обрано модуль мембранної клавіатури, що містить 12 кнопок, розташованих у вигляді матричного поля 3x4. На рисунку 2.14 зображено зовнішній вигляд цієї клавіатури.



Рисунок 2.14 – Зовнішній вигляд клавіатурного модуля з 12 кнопками

Клавіатурний модуль має сім виводів, які підключаються до мікроконтролера: три з них використовуються для послідовного сканування стовпців, а решта чотири – для зчитування стану рядків. Такий підхід дозволяє виявляти натиснення будь-якої кнопки, використовуючи мінімальну кількість портів на контролері.

## **2.3 Проєктування електричної принципової схеми системи побутової охоронної сигналізації**

### **2.3.1 Обґрунтування вибору середовища проєктування електричних схем**

Для створення електронної схеми побутової охоронної сигналізації обрано програмне середовище EasyEDA, що є хмарною багато-платформною системою для розробки електроніки. Цей застосунок дозволяє організувати повний цикл проєктування, починаючи від схеми з'єднань і закінчуючи підготовкою файлів для виготовлення друкованої плати.

До основного функціоналу EasyEDA входять такі можливості:

- 1) редактор електричних схем, що дозволяє створювати принципові з'єднання компонентів;
- 2) модуль проєктування власних електронних елементів;
- 3) редактор для розведення друкованих плат;
- 4) інтегроване хмарне сховище для зберігання даних;
- 5) інструменти керування проєктами;
- 6) симулятор роботи схем;
- 7) сервіси для замовлення виготовлення плат безпосередньо з інтерфейсу програми.

На рисунку 2.15 представлений вигляд основного вікна EasyEDA, встановленого на персональному комп'ютері.

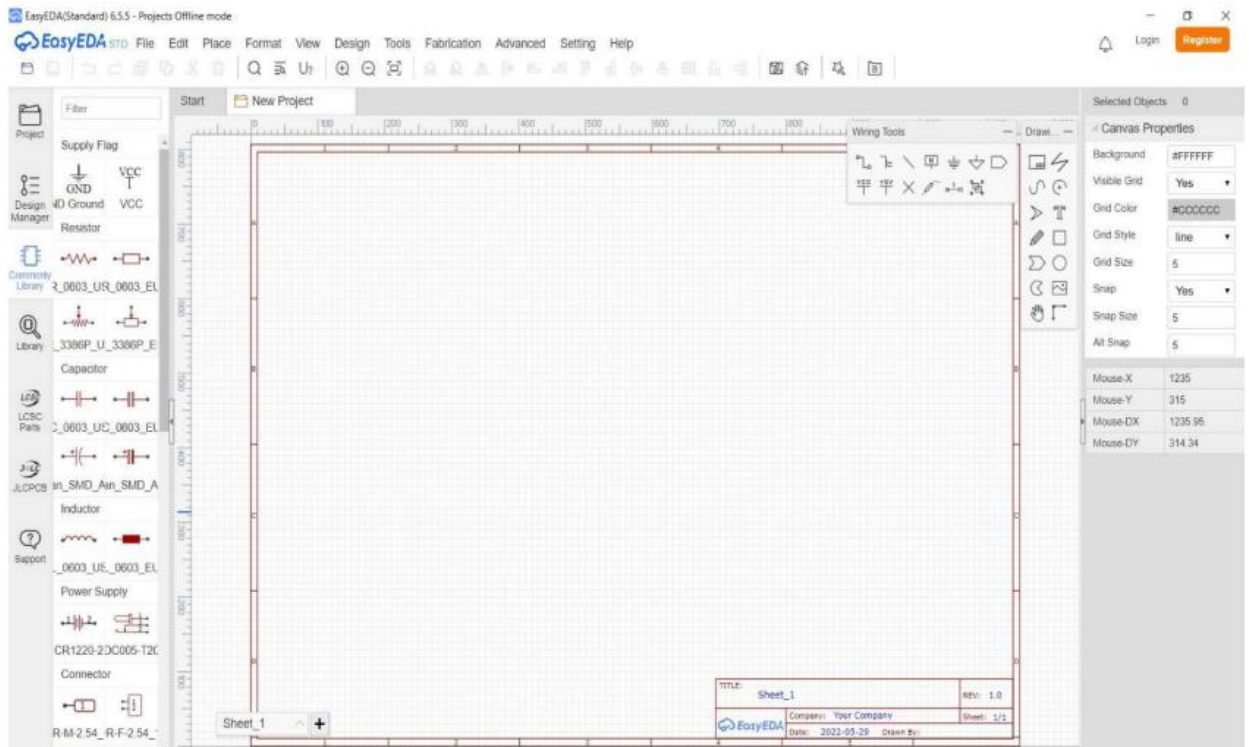


Рисунок 2.15 – Інтерфейс програми EasyEDA на ПК

Програма працює за клієнт-серверною архітектурою. Користувач може запускати інструмент безпосередньо у веб-браузері з підтримкою HTML5 або встановити локальну версію на комп'ютер. У разі використання браузера дані зберігаються у хмарі, а в локальному варіанті на жорсткому диску з можливістю подальшої синхронізації з онлайн-сховищем.

### 2.3.2 Розробка принципової схеми пристрою

На рисунку 2.16 представлена принципова електрична схема побутової охоронної сигналізації, яка створена за допомогою середовища EasyEDA.

Електроживлення у даній схемі може надходити з двох джерел: або через стандартний мережевий адаптер, підключений до роз'єму DC1, або від акумуляторної батареї В1, яка з'єднана з платою через блок захисту і контролю заряду на базі модуля U1. Оскільки модуль стільникового зв'язку SIM800L

працює при напрузі 4 В, у схемі реалізовано вузол стабілізації напруги на базі мікросхеми LM2596 (позначено як U6).

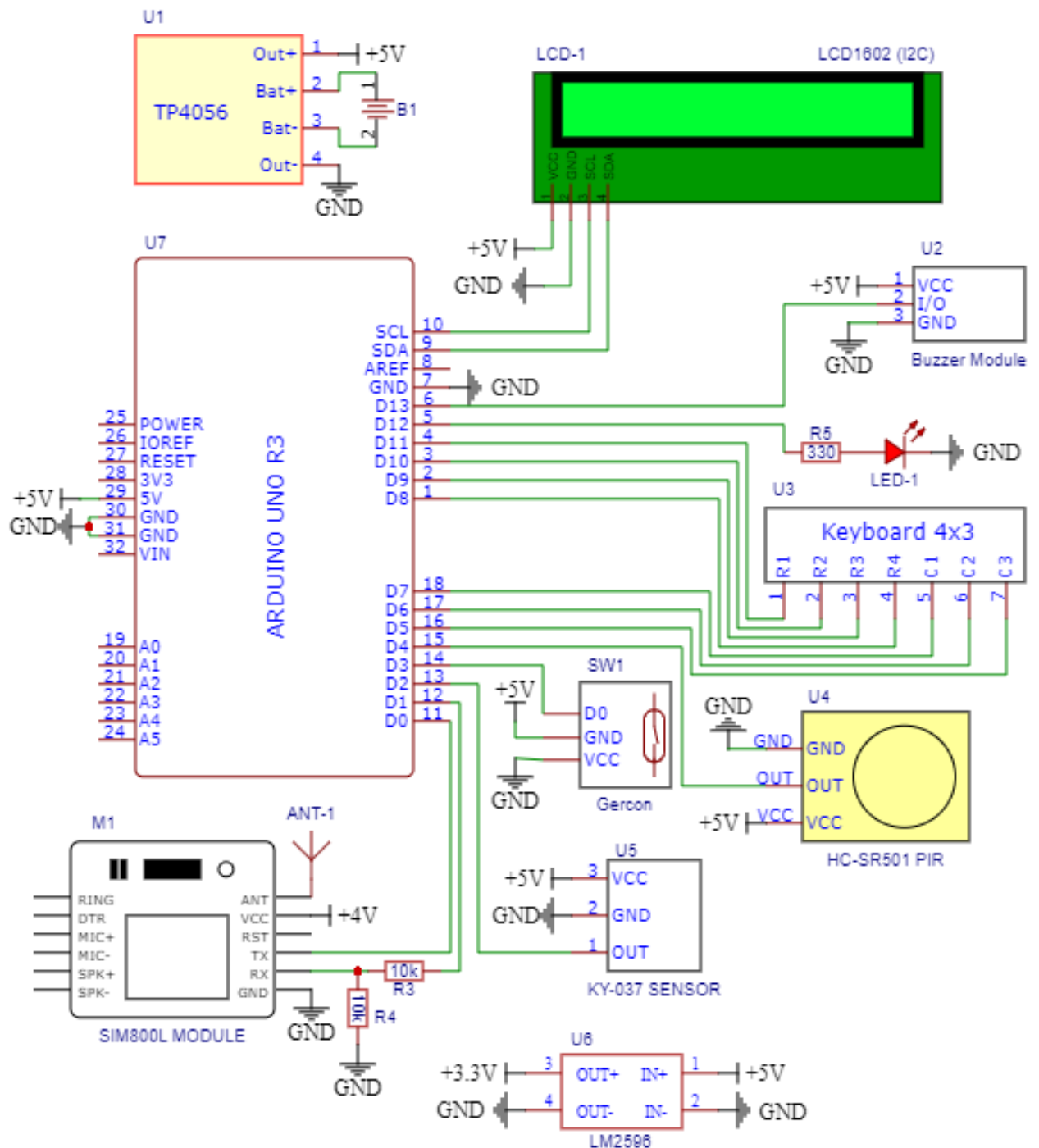


Рисунок 2.16 – Принципова схема побутової сигналізаційної системи

Датчики, що реагують на рух (U4), відчинення дверей (SW1), а також модуль, який вловлює звук розбиття скла (U5), підключені до цифрових входів мікроконтролера Arduino UNO, що на схемі позначено як U7. Матрична клавіатура (U3), яка використовується для введення коду доступу, має сім контактів: три призначені для підключення стовпців, а чотири – для рядків матриці.

Інформація з мікроконтролера на рідкокристалічний дисплей LCD-1 передається через шину I<sup>2</sup>C – лінії SDA та SCL.

Модуль стільникового зв'язку SIM800L (M1) комунікує з контролером по UART-інтерфейсу. Оскільки напруга логіки мікроконтролера відрізняється від робочої напруги GSM-модуля, між ними встановлений подільник напруги, сформований резисторами R3 і R4. Для покращення прийому сигналу використовується зовнішня антена ANT1, яка з'єднана з модулем M1.

## 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Розробка алгоритмічної моделі функціонування системи побутової охоронної сигналізації

Робота СОС починається з її ввімкнення. Після активації пристрою, контролер починає зчитувати та аналізувати сигнали, які надходять від підключених сенсорів і модулів авторизації. Мікроконтролер порівнює отримані значення з заздалегідь заданими порогоми, що відповідають нормальному стану. Ці порогові значення встановлюються під час початкового налаштування системи перед її експлуатацією.

Якщо який-небудь з сенсорів фіксує зміну, що перевищує допустимий рівень, контролер моментально розпізнає цю ситуацію як тривожну і запускає відповідні сигнальні пристрої. У поточному рішенні це реалізовано через світлодіодні індикатори та п'єзозвуковий модуль. Додатково, за допомогою GSM-модуля надсилається текстове повідомлення або здійснюється телефонний виклик на номер власника об'єкта. У разі необхідності, повідомлення також може бути передане до чергового пункту охоронної служби.

Щоб вимкнути режим охорони, користувач повинен ввести спеціальний код доступу. Після успішної авторизації дозволяється вхід у приміщення та змінення режиму функціонування системи. Водночас, сенсори продовжують працювати в звичайному режимі, однак мікроконтролер ігноруватиме сигнали тривоги від них.

Програмна частина, що завантажується в мікроконтролер, має дві основні фази. Перша фаза виконується один раз після подачі живлення, в ході якої здійснюється конфігурація портів введення-виведення та налаштування послідовного інтерфейсу. Друга частина – це основний цикл, що повторюється безперервно протягом усього часу, поки система залишається під напругою.

Блок-схемне зображення алгоритму роботи СОС подано на рисунку 3.1.

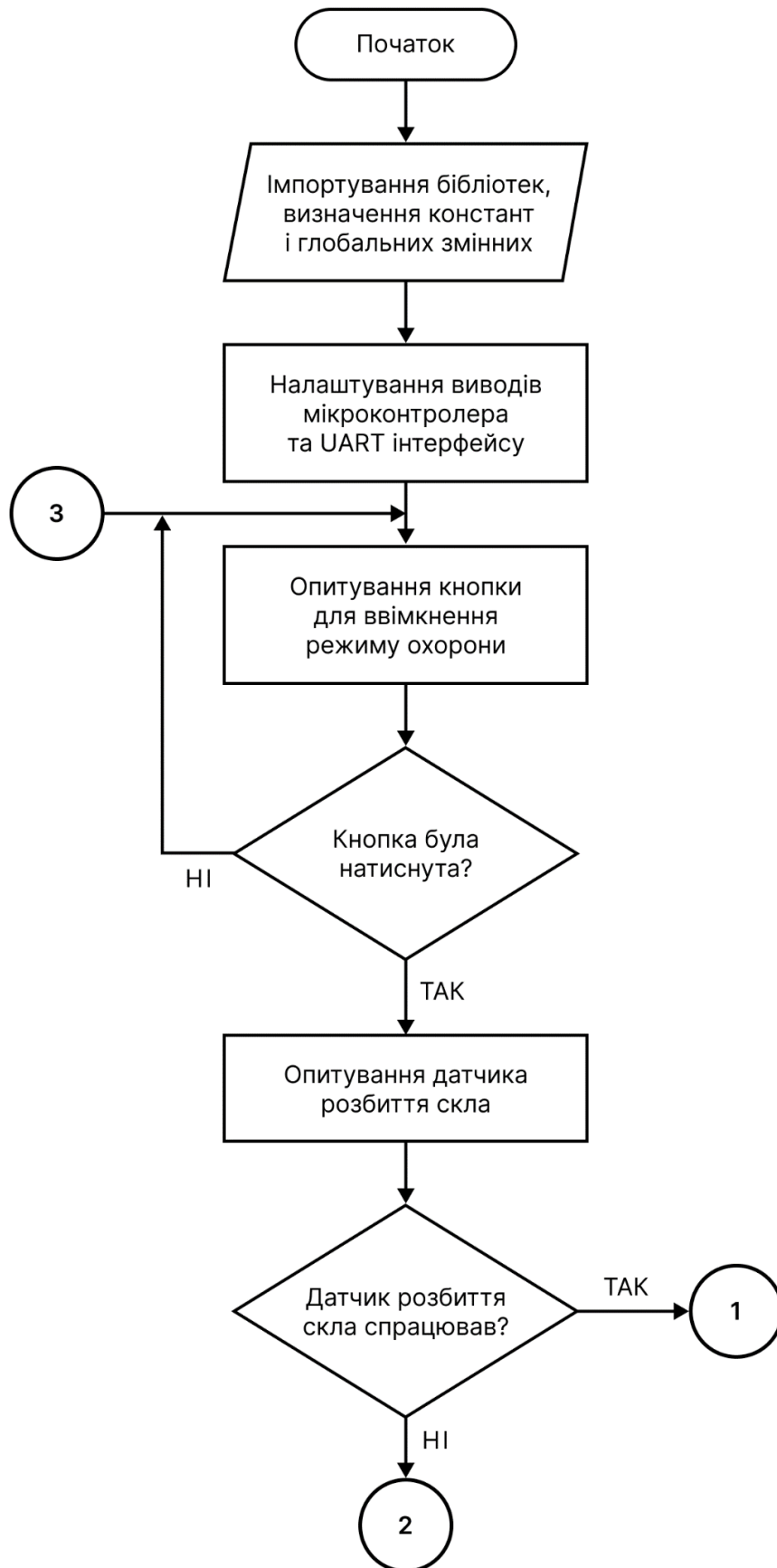


Рисунок 3.1 – Блок-схема алгоритму роботи системи охоронної сигналізації

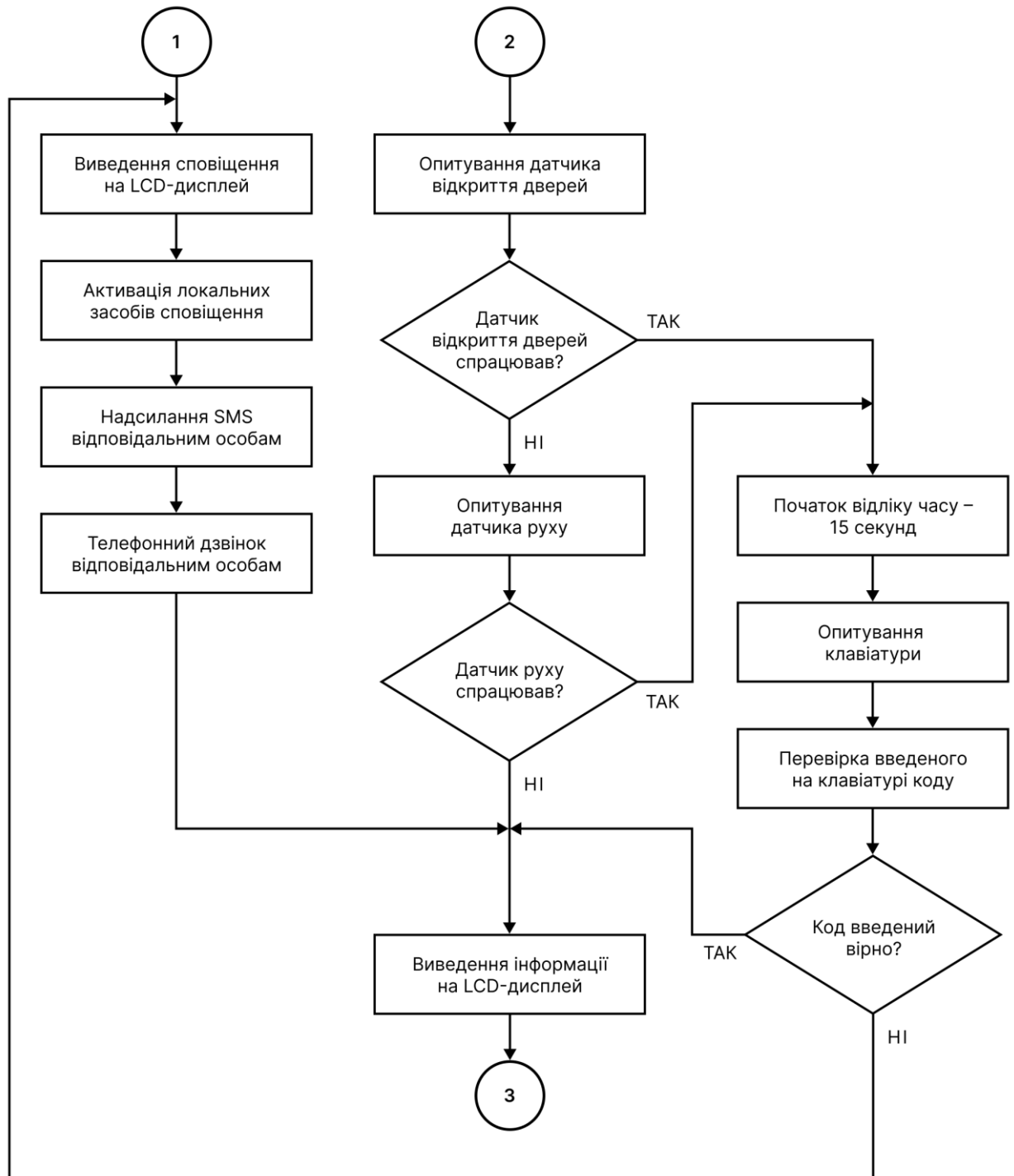


Рисунок 3.1, аркуш 2

Робота програмного забезпечення починається з підключення необхідних бібліотек та ініціалізації режимів роботи виводів мікроконтролера. Алгоритм функціонування системи реалізується через послідовність

перевірок, серед яких три стосуються аналізу даних від сенсорів, а ще дві – опитування кнопок і клавіатурного модуля.

Для підвищення зручності обслуговування та реалізації модульного підходу, структура програми розділена на наступні функціональні блоки:

- 1) модуль зчитування вхідних даних;
- 2) модуль, що відповідає за зміну режимів роботи;
- 3) блок обробки активного стану системи.

Система може функціонувати в одному з чотирьох основних режимів:

- 1) очікування – сигнали з сенсорів не обробляються, користувач може вільно перебувати у приміщенні;
- 2) охорона – усі датчики активні, будь-яке порушення фіксується;
- 3) спрацювання – якщо спрацював датчик руху чи відкриття дверей, користувач має 15 секунд для введення коду;
- 4) тривога – активується сигналізація та система сповіщення.

Основний цикл програми починається зі зчитування даних з усіх портів, до яких підключені сенсори. Зібрана інформація зберігається у відповідних змінних для подальшої обробки. Окрема увага приділяється обробці клавіатури та кнопок. Одна з кнопок використовується для переведення системи з режиму «очікування» в «охорону». Для активації охоронного режиму необхідно утримувати її протягом трьох секунд. Після цього користувачу надається 15 секунд для виходу з приміщення без ризику активації сигналізації.

Першою виконується перевірка сенсора на розбиття скла. У разі його спрацювання, система миттєво переходить у режим «тривоги». Далі обробляються дані від сенсорів руху та відкриття. Якщо хоча б один із них активується, запускається режим «спрацювання».

Для зручності налагодження передбачена функція виведення змінних у послідовний порт, що дозволяє в режимі реального часу відстежувати зміни в роботі системи.

Блок обробки режимів реалізує відповідні дії згідно з поточним станом:

- 1) охорона – виходи для світлової та звукової сигналізації вимкнені;
- 2) спрацювання – запускається таймер очікування введення правильного коду, інакше система змінить режим роботи на «тривога»;
- 3) тривога – активуються засоби оповіщення, GSM-модуль надсилає SMS та виконує виклик на заданий номер.

Після завершення обробки всіх умов керування передається на початок циклу. Це забезпечує постійний контроль над безпекою об'єкта з боку охоронної системи.

У наступному підрозділі буде здійснено детальний аналіз програмного коду, який реалізує описаний алгоритм.

## **3.2 Налаштування середовища для розробки ПЗ**

### **3.2.1 Середовище розробки програмного коду для мікроконтролера**

Для створення програмного забезпечення мікроконтролерів, що використовуються у платформах типу Arduino, застосовується мова програмування Processing, яка є спрощеним варіантом C/C++ з розширеним набором бібліотек. Для написання програмного коду в даному проєкті використано Arduino IDE – кросплатформенне середовище розробки, реалізоване на основі мови Java. На рисунку 3.3 представлено зовнішній вигляд головного вікна цього середовища.

Arduino IDE об'єднує в собі кілька ключових компонентів:

- 1) вікно редагування коду;
- 2) компілятор, який здійснює перевірку та трансляцію скетчів;
- 3) інструмент для завантаження прошивки безпосередньо до мікроконтролера через відповідний інтерфейс.

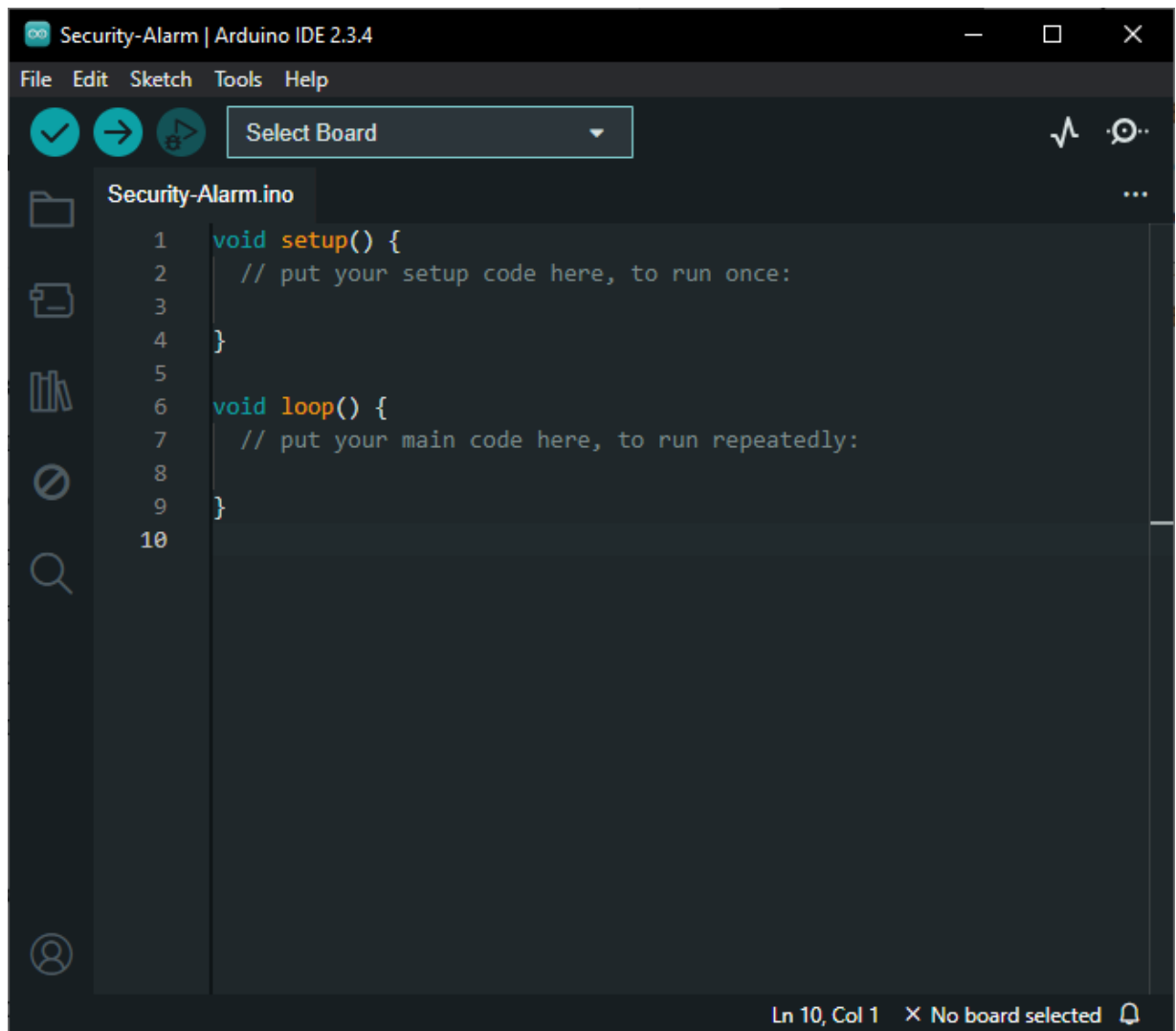


Рисунок 3.3 – Інтерфейс середовища розробки Arduino IDE

### 3.2.2 Підключення бібліотеки для взаємодії з GSM-модулем

Щоб забезпечити коректну взаємодію мікроконтролера з GSM-модулем SIM800L, у середовищі Arduino IDE потрібно попередньо інсталиювати відповідну бібліотеку. Для цього користувач переходить до розділу «Tools» та обирає опцію «Manage Libraries». У вікні бібліотек, що відкривається (Library Manager), у пошуковому рядку вказується назва модуля – «SIM800L». Після відображення результатів пошуку залишається лише натиснути кнопку «Install» напроти відповідної бібліотеки, як це показано на рисунку 3.4.

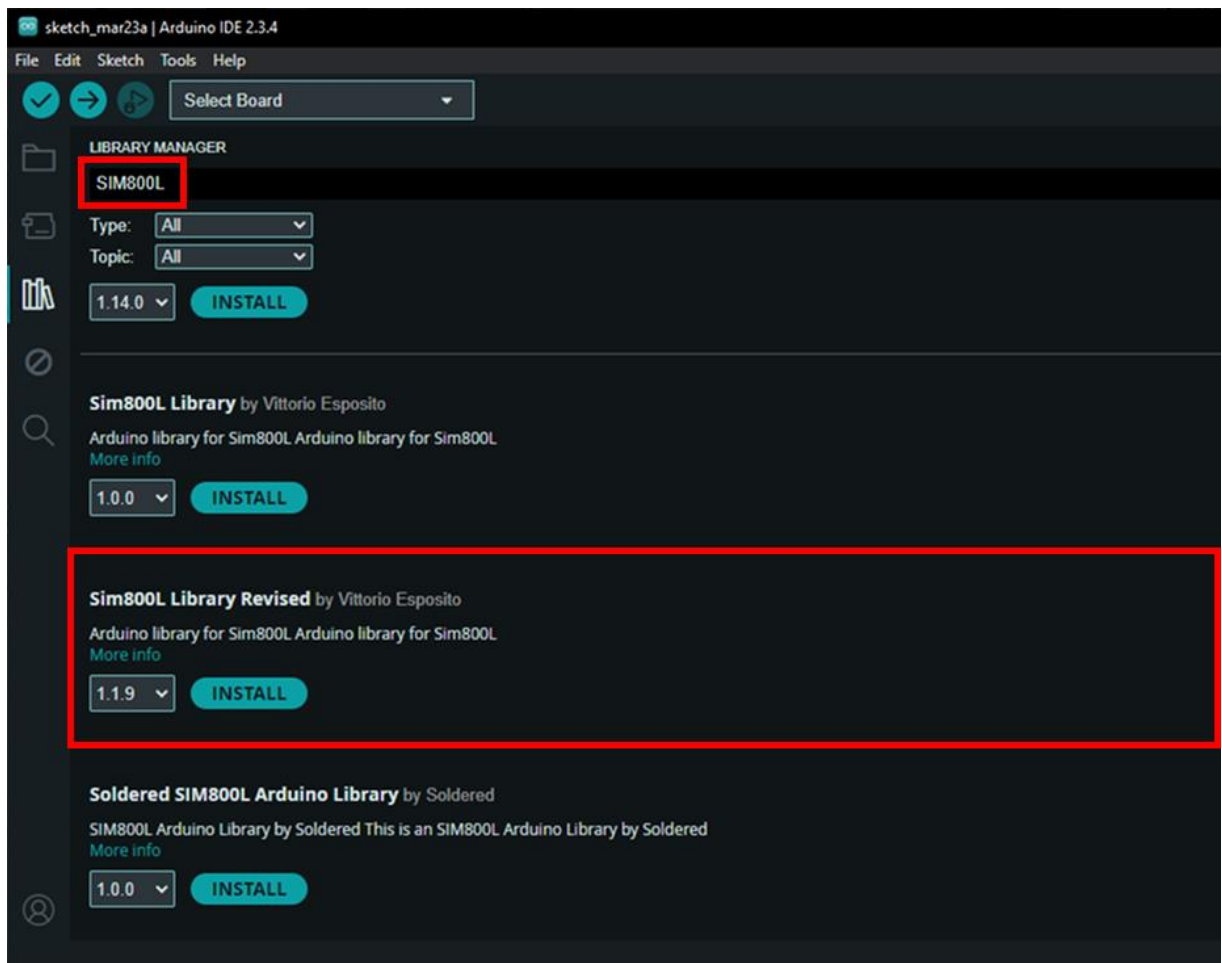


Рисунок 3.4 – Процес встановлення бібліотеки для роботи з модулем SIM800L в Arduino IDE

### 3.3 Реалізація ПЗ системи побутової охоронної сигналізації

У цьому підрозділі подано опис основних програмних функцій і фрагментів коду, що реалізують функціональні можливості системи побутової охоронної сигналізації безпосередньо на рівні мікроконтролера.

#### 3.3.1 Код для опитування клавіатури

Для реалізації роботи з клавіатурою 3x4 у програмному коді визначено константи, які відповідають пінам мікроконтролера Arduino, підключеним до

рядків і стовпців клавіатури. Також створено двовимірний масив символів  $k3 \times 4$ , який відображає розкладку кнопок клавіатури (лістинг 3.1).

```
// Піни для клавіатури
const int Row[] = {11, 10, 9, 8}; // Рядки
const int Col[] = {7, 6, 5};      // Стовпці

// Символи клавіатури 3x4
const char k3x4[4][3] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}};
```

Лістинг 3.1 – Визначення констант для роботи з клавіатурою

У функції `setup()` ініціалізуються виводи мікроконтролера для роботи з клавіатурою (лістинг 3.2). Виводи, підключені до рядків (`Row`), налаштовуються як цифрові виходи з високим рівнем напруги (`HIGH`), а виводи, підключені до стовпців (`Col`), – як цифрові входи з підтяжкою до рівня логічної одиниці (`INPUT_PULLUP`). Це забезпечує коректне опитування стану кнопок.

```
void setup() {
  // Налаштування клавіатури
  for (int i = 0; i < 4; i++) {
    pinMode(Row[i], OUTPUT);
    digitalWrite(Row[i], HIGH);
  }
  for (int i = 0; i < 3; i++) {
    pinMode(Col[i], INPUT_PULLUP);
  }
}
```

Лістинг 3.2 – Налаштування виводів для роботи з клавіатурою

У функції `loop()` реалізовано опитування клавіатури (лістинг 3.3). Для цього в циклі послідовно на виводи рядків подається низький рівень напруги (`LOW`), а у вкладеному циклі перевіряється стан виводів стовпців. Якщо на стовпці виявлено рівень логічного "0", це означає, що рядок замкнутий зі

стовпцем, і відповідна кнопка натиснута. Функція `checkKeyState()` повертає символ натиснутої кнопки, який визначений у масиві `k3x4`.

```
char checkKeyState() {
    for (int row = 0; row < 4; row++) {
        digitalWrite(Row[row], LOW);
        for (int col = 0; col < 3; col++) {
            if (digitalRead(Col[col]) == LOW) {
                char key = k3x4[row][col];
                digitalWrite(Row[row], HIGH);
                return key;
            }
        }
        digitalWrite(Row[row], HIGH);
    }
    return 0;
}
```

Лістинг 3.3 – Функція опитування клавіатури

### 3.3.2 Код для виведення інформації на LCD-дисплей

Для роботи з LCD-дисплеєм у програмі підключено бібліотеку `LiquidCrystal_I2C.h`, яка забезпечує взаємодію з дисплеєм через інтерфейс I<sup>2</sup>C. Об'єкт дисплея ініціалізується з адресою I<sup>2</sup>C `0x27` та розмірами `16x2` (16 символів у ширину та 2 рядки) (лістинг 3.4).

```
#include <LiquidCrystal_I2C.h>

// Ініціалізація LCD (адреса 0x27, 16x2)
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Лістинг 3.4 – Ініціалізація LCD-дисплея

У функції `setup()` здійснюється ініціалізація дисплея, увімкнення підсвітки та виведення початкового повідомлення "Waiting mode" разом із інструкцією "Enter \*# to arm" (лістинг 3.5).

```

void setup() {
  // Ініціалізація LCD
  lcd.init();
  lcd.backlight();
  displayWaitingMode();
}

void displayWaitingMode() {
  lcd.clear();
  lcd.print("Waiting mode");
  lcd.setCursor(0, 1);
  lcd.print("Enter *# to arm");
}

```

Лістинг 3.5 – Ініціалізація та виведення початкового повідомлення на LCD

### 3.3.3 Код для обміну даними з GSM-модулем

Для обміну даними з GSM-модулем SIM800L використовується інтерфейс UART, реалізований через бібліотеку `SoftwareSerial.h`. Це дозволяє мікроконтролеру Arduino спілкуватися з модулем через програмний послідовний порт на пінах D0 (RX) і D1 (TX) (лістинг 3.6).

```

#include <SoftwareSerial.h>

// Створюємо об'єкт для зв'язку з SIM800L (RX, TX)
SoftwareSerial sim800l(0, 1); // D0 (RX) і D1 (TX) для SIM800L

void setup() {
  // Ініціалізація SIM800L
  sim800l.begin(9600); // Запускаємо зв'язок з SIM800L на
швидкості 9600 бод
  delay(1000); // Даємо модулю час на ініціалізацію
  initializeGSM(); // Ініціалізуємо GSM-модуль
}

```

Лістинг 3.6 – Ініціалізація UART для обміну даними з GSM-модулем

Функція `initializeGSM()` надсилає команди AT для перевірки зв'язку з модулем і налаштування текстового режиму для SMS (лістинг 3.7). Відповіді від модуля зчитуються через `sim800l.available()` і виводяться в `Serial Monitor` за допомогою `sim800l.read()` та `Serial.write()`.

```

void initializeGSM() {
  Serial.println("Initializing GSM module...");
  sim8001.println("AT");          // Перевіряємо зв'язок з модулем
  delay(1000);
  sim8001.println("AT+CMGF=1"); // Встановлюємо текстовий режим
  для SMS
  delay(1000);
  while (sim8001.available()) {
    Serial.write(sim8001.read());
  }
}

```

### Лістинг 3.7 – Надсилання та отримання даних через UART

Функція `sendSMS()` відповідає за відправку SMS-повідомлення через GSM-модуль (лістинг 3.8). Вона формує повідомлення залежно від типу тривоги (рух, двері, розбиття скла) і надсилає його на визначений номер телефону.

```

void sendSMS() {
  String message = "";
  if (motionDetected == 1) {
    message = "Motion: Alarm!";
  } else if (doorDetected == 1) {
    message = "Door: Alarm!";
  } else if (glassDetected == 1) {
    message = "Glass: Alarm!";
  }

  Serial.println("Sending SMS...");
  sim8001.println("AT+CMGF=1"); // Встановлюємо текстовий режим
  SMS
  delay(100);
  sim8001.println("AT+CMGS=\"\" + phoneNumber + "\""); //
  Вказуємо номер телефону
  delay(100);
  sim8001.print(message); // Відправляємо текст повідомлення
  delay(100);
  sim8001.write(26);
  delay(100);
  Serial.println("SMS Sent.");
}

```

### Лістинг 3.8 – Функція для надсилання SMS через GSM-модуль

Повний текст коду СОС наведено у додатку А.

## ВИСНОВКИ

У кваліфікаційній роботі вирішено практичне завдання – створення побутової системи охорони житлових приміщень. У ході розробки вдалося досягти наступних результатів:

1) проведено огляд сучасних рішень у сфері побутових охоронних систем та визначено їх ключові сильні і слабкі сторони;

2) запропоновано структуру системи, що передбачає використання GSM технології для віддаленого контролю за станом охоронних сенсорів (додаток Б);

3) розроблено електричну принципову схему керуючого модуля сигналізації;

4) описано алгоритмічну логіку функціонування системи та створено програмне забезпечення, яке реалізує всі необхідні функції згідно з технічним завданням.

Розроблена КС побутової охоронної системи зібрана у вигляді демонстраційного макету, який пройшов тестування. Перевірка працездатності показала, що всі елементи працюють стабільно та відповідають заданим технічним характеристикам. Запропонована система може бути ефективно застосована для охорони житлових об'єктів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Квартирні крадіжки: у 2024 році. [Електронний ресурс] – Режим доступу: [https://skilky-skilky.info/kvartyrni-kradizhky-u-2024-rotsi-kilkist-kryminalnykh-provadzhen-siahnula-minimumu-1914-sprav/?utm\\_source=chatgpt.com](https://skilky-skilky.info/kvartyrni-kradizhky-u-2024-rotsi-kilkist-kryminalnykh-provadzhen-siahnula-minimumu-1914-sprav/?utm_source=chatgpt.com) (дата звернення: 08.03.2025)
2. Михальчук Д. О., Яворська О. М. Аналіз ринку систем охоронної сигналізації. Матеріали 75-ї науково-технічної конференції професорсько-викладацького складу, науковців, аспірантів та студентів. 2020. С. 49-50.
3. Ahmad M. B., Abdullahi A. A., Muhammad A. S., Saleh Y. B., Usman U. B. The Various Types of sensors used in the Security Alarm system. International Journal of New Computer Architectures and their Applications (IJNCAA). 2019. 9(2). P. 50-59.
4. Погребенник В. Д., Політило Р. В. Ультразвукові сенсори системи охоронної сигналізації. Вісник НТУУ “КПІ”. Серія «Приладобудування». 2008. Вип. 36. С. 68-76.
5. Кугір А. В. Автоматизована система охоронної сигналізації для промислового підприємства / А. В. Кугір; наук. керівник доц. К. Л. Хрустальов // Радіоелектроніка та молодь у XXI столітті: матеріали 25-го Міжнар. молодіж. форуму, 20 квітня 2021 р. – Харків: ХНУРЕ, 2021. – Т. 2. – С. 75–76
6. Çavaş M., Ahmad M. B. A review advancement of security alarm system using internet of things (IoT). International Journal of New Computer Architectures and their Applications (IJNCAA). 9 (2). 2019.P. 38-49.
7. Кучеров Д. П., Березкін А. Л. Радіоканал LORA в системі охоронної сигналізації. Наукоємні технології. 2019. № 3(43). С. 357-363.
8. Комплект бездротової сигналізації MAKS PRO Black. [Електронний ресурс] – Режим доступу: <https://securitylab.com.ua/ua/maks-pro-black/> (дата звернення: 01.04.2025).

9. Комплект бездротової сигналізації ATIS Kit GSM 100. [Електронний ресурс] – Режим доступу: <https://securitylab.com.ua/ua/atis-kit-gsm-100/> (дата звернення: 01.04.2025).

10. StarterKit – стартовий комплект системи безпеки Ajax. [Електронний ресурс] – Режим доступу: <https://ajax.systems.ua/products/starterkit/> (дата звернення: 01.04.2025).

11. Бездротова кімнатна сирена Ajax HomeSiren. [Електронний ресурс] – Режим доступу: <https://ajax.systems.ua/products/homesiren/> (дата звернення: 01.04.2025).

12. Таранюк О.А., Шугайло Ю.Б. Огляд комп'ютерної системи охоронної сигналізації // Інформатика, інформаційні системи та технології: тези доповідей XXII Всеукраїнської конференції студентів і молодих науковців. – Одеса, 25 квітня 2025 р. – Одеса, 2025. С. 59–261.

13. Arduino Uno R3. [Електронний ресурс] – Режим доступу: <https://www.mini-tech.com.ua/ua/arduino-uno> (дата звернення: 18.04.2025).

14. Arduino UNO – популярна плата розробки. [Електронний ресурс] – Режим доступу: <https://itmaster.biz.ua/directory/kits-nabory/arduino-uno.html> (дата звернення: 18.04.2025).

15. Датчик руху (PIR Motion sensor) HC-SR501 . [Електронний ресурс] – Режим доступу: <https://radiostore.ua/products/datchik-dvizheniya-pir-motion-sensor-hc-sr501-2> (дата звернення: 18.04.2025).

16. Магнітно-контактний герконовий датчик відкриття дверей МС-38. [Електронний ресурс] – Режим доступу: <https://diyshop.com.ua/ua/magnitno-kontaktnyj-gerkonovyj-datchik-otkrytiya-dveri-okna-mc-38?srsId=AfmBOoG3F4ibjIPJFICt49cr9h68cJ69ANE1HIxb7VuTsG1LHIKj6F2> (дата звернення: 18.04.2025).

17. Модуль мікрофон Arduino із високою чутливістю KY-037 (датчик звуку) . [Електронний ресурс] – Режим доступу: <https://mrobot.com.ua/product/modul-mikrofon-arduino-iz-vysokoyu-chutlyvistyuu-ky-037-datchyk-zvuku/?srsId=AfmBOoq6FiV4Uv-->

PHZS2g2fN5dCHTsN7Cwj\_PYLmKA-NX6mJSaL0KmG (дата звернення: 08.05.2025).

18. GSM-модуль на SIM800L. [Електронний ресурс] – Режим доступу: <https://arduino.ua/prod1665-gsm-modyl-nasim800l> (дата звернення: 08.05.2025).

19. Зарядний модуль TP4056 Type-C з функцією захисту акумулятора. [Електронний ресурс] – Режим доступу: <https://arduino.ua/prod4517-zaryadnii-modyl-tp4056-type-c-s-funkcieizashhiti-akkumulyatora> (дата звернення: 08.05.2025).

20. Модуль п'єзодинаміка. [Електронний ресурс] – Режим доступу: <https://uamper.com/Buzzer-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8C-%D0%B4%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D0%BA%D0%BE%D0%BC-%D0%BF%D1%8C%D0%B5%D0%B7%D0%BE-%D0%BF%D0%B8%D1%89%D0%B0%D0%BB%D0%BA%D0%B0> (дата звернення: 08.05.2025).

21. LCD дисплей 1602 із зеленим підсвічуванням. [Електронний ресурс] – Режим доступу: <https://ardushop.in.ua/arduino/lcd-display-1602-hd44780-with-green-backlight> (дата звернення: 08.05.2025).

22. I<sup>2</sup>C модуль розширення виводів Arduino для підключення LCD дисплея. [Електронний ресурс] – Режим доступу: <https://arduino.ua/prod1790-iici2cinterfeis-lcd1602-2004> (дата звернення: 08.05.2025).

## ДОДАТОК А

### Лістинг програми

```

#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Створюємо об'єкт для зв'язку з SIM800L (RX, TX)
SoftwareSerial sim800l(0, 1); // D0 (RX) і D1 (TX) для SIM800L

// Номер телефону для відправки SMS (замініть на свій)
const String phoneNumber = "+380123456789"; // Вкажіть свій
номер у форматі +код_країни_номер

// Піни для клавіатури
const int Row[] = {11, 10, 9, 8}; // Рядки
const int Col[] = {7, 6, 5}; // Стовпці

// Символи клавіатури 3x4
const char k3x4[4][3] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

// Піни для датчиків і пристроїв
const int motionSensor = 4; // Датчик руху HC-SR501
const int doorSensor = 3; // Геркон МС-38
const int glassBreakSensor = 2; // Датчик звуку/вібрації KY-037
const int buzzer = 13; // П'єзодинамік
const int ledPin = 12; // Світлодіод

// Змінні для стану системи
int sensorState = 0; // 0 - очікування, 1 - охорона,
2 - тривога, 3 - спрацьовування
int motionDetected = 0; // 0 - не виявлено, 1 -
виявлено
int doorDetected = 0; // Те саме для дверей
int glassDetected = 0; // Те саме для скла
unsigned long lastEventTime = 0; // Час останньої події
const long delayInterval = 15000; // 15 секунди затримки
unsigned long exitDelayStart = 0; // Час початку затримки виходу
const long exitDelay = 15000; // 15 секунди для виходу
String inputString = ""; // Рядок для введення
String password = "0803#"; // Поточний пароль

```

Лістинг А.1 – Програма для мікроконтролера для реалізації процесу керування системою побутової охоронної сигналізації

```

String armingCode = "*#";           // Код активації
String newPassword = "";            // Змінна для нового пароля
bool changingPassword = false;     // Флаг для режиму зміни пароля
int displayStartPos = 0;           // Позиція початку відображення
для прокрутки пароля

// Змінні для сирени
unsigned long sirenStartTime = 0;
const long sirenDuration = 30000; // 30 секунд
unsigned long lastToneChange = 0; // Час останньої зміни тону
bool buzzerState = false;         // Стан бузера (HIGH/LOW)
int tonePhase = 0;                // 0 - перший тон (1 мс), 1 -
другий тон (2 мс)
const int tone1Delay = 1;         // Затримка для першого тону
(1 мс)
const int tone2Delay = 2;         // Затримка для другого тону
(2 мс)

// Змінні для клавіатури
bool keyPressed = false;          // Чи була кнопка натиснута
char lastKey = 0;                 // Остання натиснута кнопка

// Змінні для відображення "Wrong pass!"
bool showingWrongPass = false;    // Чи відображається "Wrong
pass!"
unsigned long wrongPassStartTime = 0; // Час початку
відображення "Wrong pass!"
const long wrongPassDisplayDuration = 2000; // 2 секунди
відображення

// Флаг для запобігання повторної відправки SMS
bool smsSent = false; // Флаг для відстеження, чи було SMS
відправлено

// Ініціалізація LCD (адреса 0x27, 16x2)
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
// Налаштування клавіатури
for (int i = 0; i < 4; i++) {
pinMode(Row[i], OUTPUT);
digitalWrite(Row[i], HIGH);
}
for (int i = 0; i < 3; i++) {
pinMode(Col[i], INPUT_PULLUP);
}

// Налаштування датчиків і пристроїв
pinMode(motionSensor, INPUT);
pinMode(doorSensor, INPUT_PULLUP);

```

```

pinMode(glassBreakSensor, INPUT);
pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);

// Ініціалізація LCD
lcd.init();
lcd.backlight();
displayWaitingMode();

// Ініціалізація Serial для налагодження
Serial.begin(9600);

// Ініціалізація SIM800L
sim800l.begin(9600); // Запускаємо зв'язок з SIM800L на
швидкості 9600 бод
delay(1000); // Даємо модулю час на ініціалізацію
initializeGSM(); // Ініціалізуємо GSM-модуль
}

void loop() {
// Обробка введення з клавіатури
char key = checkKeyState();
if (key && !keyPressed) { // Обробляємо натискання лише раз
keyPressed = true;
lastKey = key;
inputString += key;

// Відображення введення залежно від стану
if (!showingWrongPass) {
if (changingPassword) {
if (key != '#') { // Не показуємо "#" під час введення нового
пароля
newPassword += key;
}
}
lcd.setCursor(0, 1);
lcd.print(newPassword + "   "); // Очищаємо зайві символи
} else if (sensorState == 2 || sensorState == 3) {
// Прокрутка пароля, якщо він довший за 5 символів
int passLength = inputString.length();
int maxDisplayChars = 5; // Доступне місце після "Enter pass: "
if (passLength <= maxDisplayChars) {
displayStartPos = 0;
} else {
displayStartPos = passLength - maxDisplayChars;
}

// Відображаємо видиму частину пароля
lcd.setCursor(11, 1); // Позиція після "Enter pass: " (11
символів)
String visiblePart = inputString.substring(displayStartPos);

```

```

lcd.print(visiblePart + "      "); // Додаємо пробіли для
очищення
} else {
lcd.setCursor(0, 1);
lcd.print(inputString + "      ");
}
}

// Активация режиму охорони
if (sensorState == 0 && inputString.endsWith(armingCode) &&
!changingPassword) {
sensorState = 1;
exitDelayStart = millis();
lcd.clear();
lcd.print("Armed - Exit 15s");
digitalWrite(ledPin, HIGH);
inputString = "";
displayStartPos = 0;
}

// Перевірка пароля в режимі тривоги або спрацьовування
if ((sensorState == 2 || sensorState == 3) &&
inputString.endsWith("#")) {
if (inputString == password) {
lcd.clear();
lcd.print("Alarm off");
stopSiren();
sensorState = 0; // Перехід у режим очікування
resetSystemState();
displayWaitingMode();
} else {
showingWrongPass = true;
wrongPassStartTime = millis();
lcd.clear();
lcd.print("Wrong pass!");
inputString = "";
displayStartPos = 0;
}
}

// Зміна пароля в режимі очікування
if (sensorState == 0 && inputString == password &&
!changingPassword) {
changingPassword = true;
newPassword = "";
lcd.clear();
lcd.print("Enter new pass:");
inputString = "";
displayStartPos = 0;
}
}

```

```

// Встановлення нового пароля
if (changingPassword && inputString.endsWith("#")) {
if (newPassword.length() > 0) { // Переконаємося, що новий
пароль не порожній
password = newPassword + "#";
changingPassword = false;
lcd.clear();
lcd.print("Pass changed!");
delay(2000); // Коротка затримка для відображення
resetSystemState();
displayWaitingMode();
} else {
changingPassword = false;
lcd.clear();
lcd.print("Pass too short!");
delay(2000);
resetSystemState();
displayWaitingMode();
}
}
}

// Скидаємо стан натискання, коли кнопка відпущена
bool anyKeyDown = false;
for (int row = 0; row < 4; row++) {
digitalWrite(Row[row], LOW);
for (int col = 0; col < 3; col++) {
if (digitalRead(Col[col]) == LOW) {
anyKeyDown = true;
break;
}
}
digitalWrite(Row[row], HIGH);
if (anyKeyDown) break;
}
if (!anyKeyDown) {
keyPressed = false;
}
}

// Логіка для різних станів
switch (sensorState) {
case 0: // Режим очікування
if (!showingWrongPass && !changingPassword) {
lcd.setCursor(0, 1);
lcd.print("Enter *# to arm");
}
break;

case 1: // Режим охорони
if (millis() - exitDelayStart < exitDelay) {

```

```

lcd.setCursor(0, 1);
lcd.print(String((exitDelay - (millis() - exitDelayStart)) /
1000) + "s left");
} else {
lcd.clear();
lcd.print("Full armed");
checkSensors();
}
break;

case 2: // Режим тривоги
if (millis() - lastEventTime >= delayInterval) {
sensorState = 3; // Перехід у режим спрацьовування
startSiren();
lcd.clear(); // Очищаємо екран перед оновленням
displayAlarmMessage(); // Оновлюємо на "Alarm!"
if (!smsSent) { // Перевіряємо, чи не було SMS вже відправлено
sendSMS(); // Відправляємо SMS
smsSent = true; // Встановлюємо флаг, щоб не відправляти
повторно
}
Serial.print("Transition to sensorState 3 at: ");
Serial.println(millis());
}
break;

case 3: // Режим спрацьовування
if (millis() - sirenStartTime < sirenDuration) {
activateBuzzer(); // Безперервна робота сирени
} else {
stopSiren();
sensorState = 2; // Повернення в режим тривоги для введення
пароля
lcd.clear(); // Очищаємо екран перед поверненням
displayAlarmMessage(); // Відображаємо оновлене повідомлення
}
break;
}

// Управління відображенням "Wrong pass!"
if (showingWrongPass && (millis() - wrongPassStartTime >=
wrongPassDisplayDuration)) {
showingWrongPass = false;
restoreDisplayAfterWrongPass();
}
}

// Функція для ініціалізації GSM-модуля
void initializeGSM() {
Serial.println("Initializing GSM module...");
}

```

```

sim8001.println("AT"); // Перевіряємо зв'язок з модулем
delay(1000);
sim8001.println("AT+CMGF=1"); // Встановлюємо текстовий режим
для SMS
delay(1000);
while (sim8001.available()) {
Serial.write(sim8001.read());
}
}

// Функція для відправки SMS
void sendSMS() {
String message = "";
if (motionDetected == 1) {
message = "Motion: Alarm!";
} else if (doorDetected == 1) {
message = "Door: Alarm!";
} else if (glassDetected == 1) {
message = "Glass: Alarm!";
}

Serial.println("Sending SMS...");
sim8001.println("AT+CMGF=1"); // Встановлюємо текстовий режим
SMS
delay(100);
sim8001.println("AT+CMGS=\"" + phoneNumber + "\""); // Вказуємо
номер телефону
delay(100);
sim8001.print(message); // Відправляємо текст повідомлення
delay(100);
sim8001.write(26); // Відправляємо Ctrl+Z (код 26) для
завершення повідомлення
delay(100);
Serial.println("SMS Sent.");
}

// Функція для відображення "Waiting mode"
void displayWaitingMode() {
lcd.clear();
lcd.print("Waiting mode");
lcd.setCursor(0, 1);
lcd.print("Enter *# to arm");
}

// Функція для скидання прапорців і змінних
void resetSystemState() {
motionDetected = 0;
doorDetected = 0;
glassDetected = 0;
inputString = "";
}

```

```

newPassword = "";
displayStartPos = 0;
smsSent = false; // Скидаємо флаг відправки SMS
}

// Функція для відновлення дисплея після "Wrong pass!"
void restoreDisplayAfterWrongPass() {
  lcd.clear();
  if (sensorState == 2 || sensorState == 3) {
    displayAlarmMessage();
    lcd.setCursor(0, 1);
    lcd.print("Enter pass: ");
  } else {
    displayWaitingMode();
  }
}

// Перевірка датчиків
void checkSensors() {
  // Перевірка датчика розбиття скла
  if (digitalRead(glassBreakSensor) == HIGH && glassDetected == 0)
  {
    glassDetected = 1;
    sensorState = 3;
    lastEventTime = millis();
    lcd.clear();
    lcd.print("Glass: Broken!");
    startSiren();
    Serial.print("Glass detected at: ");
    Serial.println(lastEventTime);
  }

  // Перевірка датчиків руху та дверей
  if (digitalRead(motionSensor) == HIGH && motionDetected == 0) {
    motionDetected = 1;
    sensorState = 2;
    lastEventTime = millis();
    lcd.clear();
    lcd.print("Motion: Detect!");
    Serial.print("Motion detected at: ");
    Serial.println(lastEventTime);
  }
  if (digitalRead(doorSensor) == HIGH && doorDetected == 0) {
    doorDetected = 1;
    sensorState = 2;
    lastEventTime = millis();
    lcd.clear();
    lcd.print("Door: Opened!");
    Serial.print("Door detected at: ");
    Serial.println(lastEventTime);
  }
}

```

```

}
// Відображення повідомлення про тривогу
void displayAlarmMessage() {
if (glassDetected == 1) {
lcd.print("Glass: Alarm!");
} else if (motionDetected == 1) {
lcd.print("Motion: Alarm!");
} else if (doorDetected == 1) {
lcd.print("Door: Alarm!");
}
}
lcd.setCursor(0, 1);
lcd.print("Enter pass: ");
}

// Перевірка стану клавіатури (неблокуюча)
char checkKeyState() {
for (int row = 0; row < 4; row++) {
digitalWrite(Row[row], LOW);
for (int col = 0; col < 3; col++) {
if (digitalRead(Col[col]) == LOW) {
char key = k3x4[row][col];
digitalWrite(Row[row], HIGH);
return key; // Повертаємо символ без затримки
}
}
digitalWrite(Row[row], HIGH);
}
return 0;
}

// Активація сирени на 30 секунд
void startSiren() {
sirenStartTime = millis();
tonePhase = 0;
lastToneChange = millis();
buzzerState = false;
digitalWrite(ledPin, HIGH);
}

// Безперервна робота сирени
void activateBuzzer() {
unsigned long currentTime = millis();
unsigned long elapsedTime = currentTime - sirenStartTime;

// Визначаємо фазу тону
if (elapsedTime < 15000) { // Перші 15 секунд - перший тон
tonePhase = 0;
} else { // Другі 15 секунд - другий тон
tonePhase = 1;
}
}

```

```
// Вибираємо затримку залежно від фази
int toneDelay = (tonePhase == 0) ? tone1Delay : tone2Delay;

// Змінюємо стан бузера кожні toneDelay мілісекунд
if (currentTime - lastToneChange >= toneDelay) {
    buzzerState = !buzzerState;
    digitalWrite(buzzer, buzzerState ? HIGH : LOW);
    lastToneChange = currentTime;
}

// Зупинка сирени
void stopSiren() {
    digitalWrite(buzzer, LOW);
    digitalWrite(ledPin, LOW);
}
```

Лістинг А.1, аркуш 10

**ДОДАТОК Б**  
**Публікація**

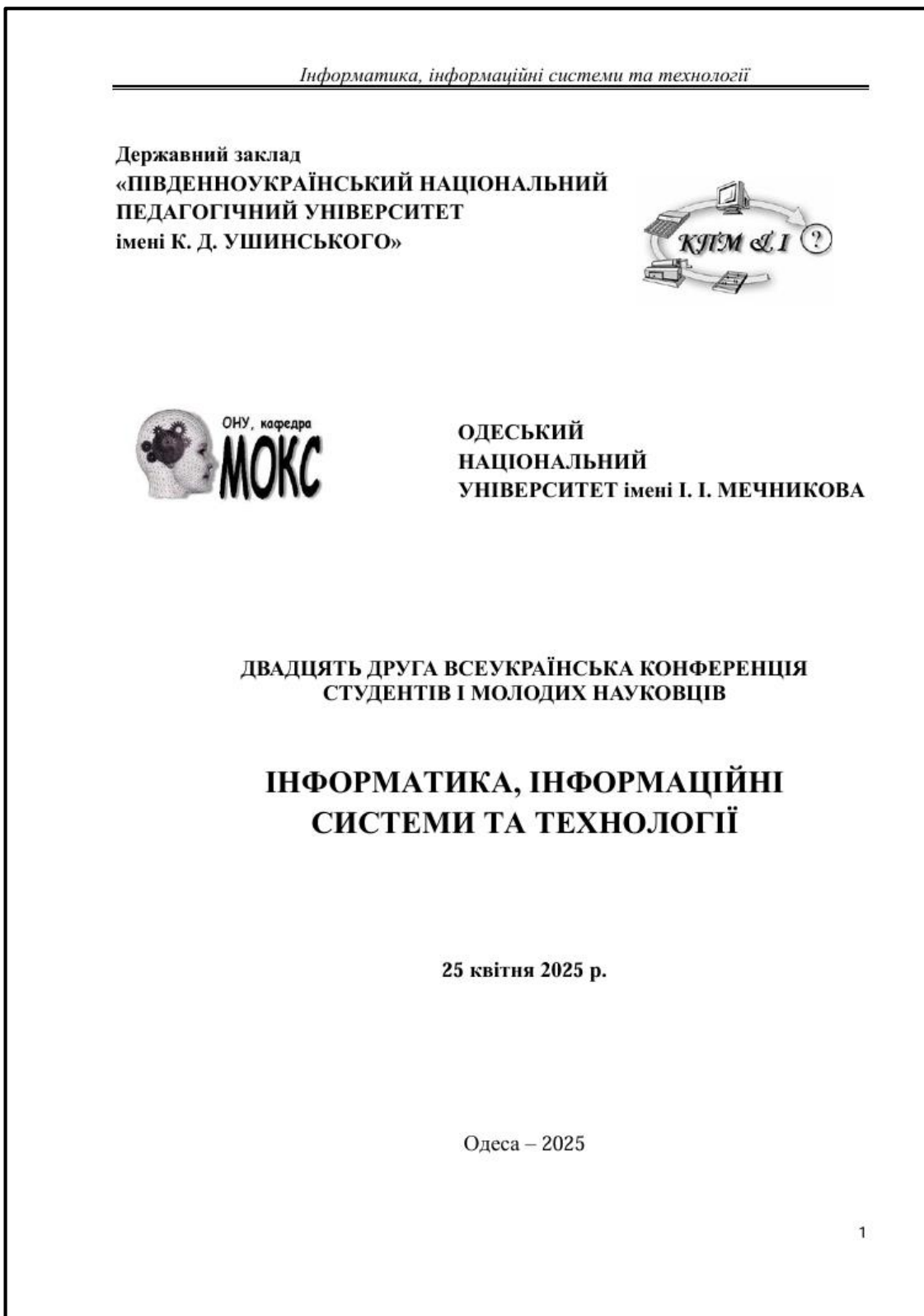


Рисунок Б.1 – Титульний лист конференції

## ОГЛЯД КОМП'ЮТЕРНОЇ СИСТЕМИ ОХОРОННОЇ СИГНАЛІЗАЦІЇ

Таранюк О.А., Шугайло Ю.Б.

Одеський національний університет імені І. І. Мечникова

*Анотація:* робота присвячена розробці комп'ютерної системи охоронної сигналізації, що дозволяє здійснювати контроль за фактом проникнення в середину приміщень.

*Ключові слова:* охоронна сигналізація, комп'ютерна система, GSM модуль, мікроконтролер, датчики руху, віддалений контроль.

Розробка комп'ютерної системи охоронної сигналізації для побутових приміщень є надзвичайно актуальним завданням, враховуючи постійне зростання рівня злочинності та необхідність оперативного реагування на спроби несанкціонованого проникнення. Запропонована система спрямована на забезпечення цілодобового контролю за станом житлових приміщень із використанням сучасних апаратно-програмних засобів, що дозволяють інтегрувати датчики, контролер та засоби сповіщення в єдиний комплекс. За основу було обрано платформу на базі мікроконтролера Arduino UNO, GSM модуль SIM800L та ряд спеціалізованих датчиків, серед яких PIR-датчик руху, датчик відкриття дверей та модуль для виявлення розбиття скла [1].

Усі датчики виділено в окремий блок. Також необхідний блок для обробки отриманих даних – блок керування. Для реалізації автономної роботи потрібно використати акумуляторну батарею з відповідним перетворювачем та системою підзарядки. Ці компоненти можна винести в окремий структурний елемент – блок живлення. Для реакції на зловмисні дії доцільно виділити окремий блок сповіщення [2].

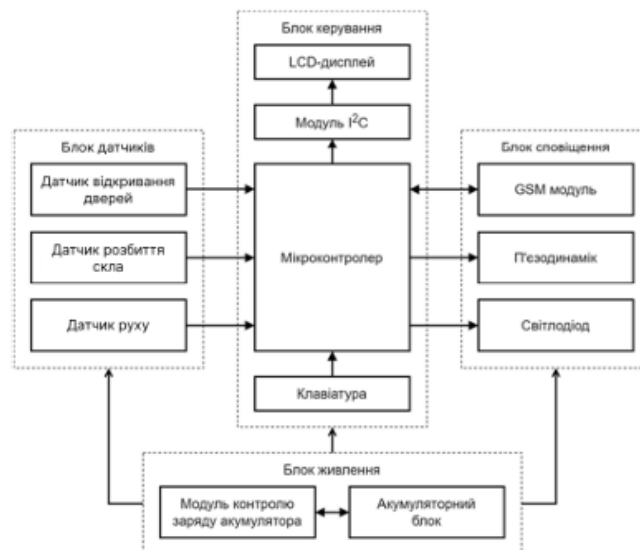


Рис. 1 – Структурна схема побутової охоронної системи

Блок датчиків призначений для виконання функцій отримання інформації про стан приміщення та її передачу в блок керування. До складу даного блоку входять такі датчики: датчик відкриття дверей, датчик розбиття скла, інфрачервоний датчик руху.

Датчик руху використовується для контролю за станом приміщення. У випадку виявлення об'єкту, який є джерелом інфрачервоного випромінювання, він спрацює і одразу передасть відповідний сигнал до блока керування.

Датчик відкриття дверей визначає факт зміни положення дверей з зачиненого стану у відчинений. Двері є одним з кількох основних шляхів зловмисного проникнення в приміщення, тому датчик, який контролює їх стан є одним з найважливіших компонентів охоронної сигналізації [3].

Датчик розбиття скла здійснює контроль за станом вікна приміщення. У випадку розбиття віконного скла, цей датчик вимірює акустичні коливання, які властиві даному явищу і надсилає відповідний сигнал до блока керування.

Отже, обраний перелік датчиків дозволить контролювати усі шляхи проникнення зловмисників в середину приміщення та фіксувати їх рух в ньому.

Блок керування містить наступні функціональні елементи: модуль плати управління на основі мікроконтролера, матрична клавіатура, модуль I<sup>2</sup>C, LCD-дисплей.

Модуль плати управління, який створений на основі мікроконтролера, є центральним елементом проекрованої системи. Він відповідає за взаємодію усіх компонентів. В якості клавіатури використовується набір з дванадцяти кнопок, які виконані у вигляді матриці 3x4, на перетині яких знаходяться кнопки. Модуль I<sup>2</sup>C, дозволяє підключити LCD-дисплей з меншою кількістю провідників.

Блок сповіщення передбачає наявність таких елементів: GSM модуль, п'єзодинамік, світлодіод.

GSM модуль використовується для передачі даних на віддалений пункт спостереження, а також для сповіщення власників приміщення через смартфон. П'єзодинамік генерує звуковий сигнал при спрацюванні сигналізації. Світлодіод застосовується для світлового сповіщення про наявність стану тривоги.

#### Література

1. Кугір А. В. Автоматизована система охоронної сигналізації для промислового підприємства. 2021. С. 75-76.
2. Лирик І.В. Комп'ютерна система побутової охоронної сигналізації: кваліфікаційна робота бакалавра за спеціальністю „123 — Комп'ютерна інженерія“ / Лирик Іван Васильович – Тернопіль, ТНТУ, 2022 – 75с.
3. Abdulkadir, Z., Ibrahim, A., Baballe, M. A., Muhammad, A., Sani, S. A. (2023). Contribution of the IoT to the Security System. J Math Techniques Comput Math, 2(9), 416-419.