

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «РОЗРОБКА GSM СИГНАЛІЗАЦІЇ  
З ВИКОРИСТАННЯМ NODEMCU»

Виконав: студент 4 курсу, групи 6.1219-2п  
спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)

освітньої програми програмна інженерія  
(назва освітньої програми)

І.А. Соколовський

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,  
доцент, к.т.н. Мухін В.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної  
математики, професор, д.т.н. Гребенюк С.М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

**ЗАТВЕРДЖУЮ**

Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

“ 07 ” 02 2023 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Соколовському Іллі Андрійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка GSM сигналізації з використанням NodeMcu

керівник роботи Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Проектування системи сигналізації.

4. Розробка системи сигналізації.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 07.02.2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.02.2023	
2.	Збір вихідних даних.	22.02.2023	
3.	Обробка методичних та теоретичних джерел.	08.03.2023	
4.	Розробка першого та другого розділу.	05.04.2023	
5.	Розробка третього розділу.	10.05.2023	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра	01.06.2023	
7.	Захист кваліфікаційної роботи.	23.06.2023	

Студент \_\_\_\_\_  
(підпис)

І.А. Соколовський  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

В.В. Мухін  
(ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка GSM сигналізації з використанням NodeMcu»: 58 с., 26 рис., 4 табл., 17 джерел, 1 додаток.

БЕЗПЕКА, ДАТЧИК, МІКРОКОНТРОЛЕР, МОДУЛЬ, СИСТЕМА СИГНАЛІЗАЦІЇ, NODEMCU, GSM.

Об'єкт дослідження – розробка системи безпеки на базі мікроконтролера NodeMcu та використання GSM технології.

Мета роботи: розробити ефективну систему безпеки з використанням мікроконтролера NodeMcu та GSM технології, яка забезпечує надійний захист об'єктів від несанкціонованого доступу та виявлення небезпеки.

Метод дослідження – аналіз, вивчення, проєктування, розробка та тестування.

В результаті виконання кваліфікаційної роботи було успішно створено систему сигналізації, яка ефективно захищає об'єкт від несанкціонованого доступу та сповіщає адміністраторів про події шляхом SMS-повідомлень.

Процес розробки системи включав дослідження предметної області, вивчення теоретичної інформації, вибір необхідних компонентів та розробку програмного коду. Використання мікроконтролера NodeMcu та GSM-модуля забезпечило стабільну роботу системи, а розроблений функціонал дозволяє зручно керувати системою сигналізації.

## SUMMARY

Bachelor's qualifying paper «Development of GSM Alarm System using NodeMcu»: 58 pages, 26 figures, 4 tables, 17 references, 1 supplement.

SECURITY, SENSOR, MICROCONTROLLER, MODULE, ALARM SYSTEM, NODEMCU, GSM.

The object of the study is development of a security system based on the NodeMcu microcontroller and the use of GSM technology.

The aim of the study is to develop an effective security system using the NodeMcu microcontroller and GSM technology, which provides reliable protection of objects against unauthorized access and detection of danger.

The methods of research – analysis, studying, design, development and testing.

As a result of the qualification work, a successful alarm system has been developed, which effectively protects the object from unauthorized access and notifies administrators about events through SMS messages.

The process of developing the system included researching the subject area, studying theoretical information, selecting necessary components, and developing software code. The use of the NodeMcu microcontroller and GSM module ensured the stable operation of the system, while the developed functionality allows for convenient control of the alarm system.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	8
1 Теоретичні відомості .....	10
1.1 Дослідження системи безпеки.....	10
1.1.1 Огляд системи безпеки.....	10
1.1.2 Складові системи безпеки.....	11
1.1.3 Принцип роботи системи безпеки.....	11
1.1.4 Типи датчиків безпеки.....	12
1.1.5 Центральна система керування .....	12
1.2 Бездротовий зв'язок та технологія GSM .....	13
1.2.1 Основи бездротового зв'язку.....	13
1.2.2 GSM технологія та її особливості .....	14
1.2.3 Застосування GSM в системах безпеки .....	15
1.3 Мікроконтролер NodeMcu .....	15
1.3.1 Огляд мікроконтролера .....	15
1.3.2 Архітектура та характеристики.....	17
1.4 Огляд інструменту розробки.....	18
1.4.1 Опис Arduino IDE.....	18
1.4.2 Встановлення середовища розробки Arduino IDE .....	19
1.4.3 Налаштування середовища розробки для плати ESP8266 .....	20
2 Проектування системи GSM сигналізації.....	23
2.1 Вимоги до системи сигналізації .....	23
2.1.1 Визначення функціональних вимог .....	23
2.1.2 Вимоги до безпеки та надійності .....	24
2.2 Вибір апаратних компонентів.....	25

2.2.1 Вибір мікроконтролера NodeMcu .....	26
2.2.2 Вибір GSM модуля для забезпечення зв'язку.....	28
2.2.3 Вибір сенсорів для виявлення порушень безпеки .....	31
2.3 Розробка архітектури системи.....	34
2.3.1 Опис загальної структури системи .....	34
2.3.2 Опис взаємодії між компонентами системи .....	35
3 Реалізація та тестування системи сигналізації.....	38
3.1 Розробка схеми підключення компонентів .....	38
3.2 Реалізація програмного забезпечення.....	40
3.2.1 Конфігурація мікроконтролера NodeMcu .....	40
3.2.2 Реалізація функції датчика руху.....	43
3.2.3 Реалізація функції перевірки стану батареї .....	44
3.2.4 Реалізація функції обробки СМС-команд .....	44
3.2.5 Реалізація функції відправки СМС модулем GSM .....	46
3.3 Тестування системи сигналізації.....	47
3.3.1 Тест-кейс.....	47
3.3.2 Контрольний список .....	48
Висновки .....	49
Перелік посилань.....	50
Додаток А.....	52

## ВСТУП

Сучасний світ характеризується постійними змінами та швидким технологічним прогресом. Забезпечення безпеки об'єктів стає надзвичайно важливим завданням у сфері охорони, будівництва, промисловості та приватного сектора. Велика увага приділяється розробці та вдосконаленню систем безпеки, які забезпечують ефективний моніторинг та захист об'єктів в режимі реального часу.

Однак, багато зі стандартних сигналізаційних систем є дорогими та складними у встановленні, особливо у випадку потреби в їх розширенні або модернізації. Також, існує потреба в дистанційному керуванні та моніторингу системи сигналізації, що може бути складним або обмеженим з використанням традиційних методів.

Метою цієї роботи є розробка GSM сигналізації з використанням NodeMcu – компактного мікроконтролеру на базі мікросхеми ESP8266, яка поєднує в собі переваги бездротового зв'язку GSM, простоту управління та налаштування за допомогою NodeMcu.

Завданнями дослідження є:

- аналіз систем безпеки та бездротового зв'язку для визначення сучасних тенденцій та технологій, що використовуються в цій галузі;
- визначення вимог до системи безпеки з використанням NodeMcu, включаючи функціональні та нефункціональні вимоги;
- проектування та реалізація системи безпеки з використанням NodeMcu, включаючи вибір апаратної конфігурації, розробку архітектури та програмного забезпечення, інтеграцію компонентів та тестування працездатності;
- оцінка ефективності розробленої системи та формулювання рекомендацій для подальшого вдосконалення та розвитку.

У цій роботі буде розглянуто розробка та реалізації GSM сигналізації з



використанням NodeMcu. Починаючи з аналізу вимог до системи, розробки архітектури, вибору необхідних компонентів та датчиків, розробки програмного забезпечення, інтеграції з мережею GSM, до тестування та налагодження системи. Будуть розглянуті питання щодо безпеки та конфіденційності передачі даних, а також можливості дистанційного керування та моніторингу системи через мобільний зв'язок.

Результатом цієї кваліфікаційної роботи буде функціональна прототипна система GSM сигналізації, яка забезпечуватиме надійний контроль та захист об'єктів. Використання NodeMcu спростить процес розробки системи, забезпечить гнучкість у налаштуванні та можливість дистанційного керування.

Ця робота має важливість у сфері розробки систем безпеки та може знайти застосування в домашніх, комерційних та інших сферах, де потрібний простий у використанні, ефективний та економічний засіб забезпечення безпеки об'єктів.

В подальшому розділі роботи будуть детально розглянуті питання, пов'язані з проектуванням та реалізацією системи GSM сигналізації з використанням NodeMcu.

# 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Дослідження системи безпеки

### 1.1.1 Огляд системи безпеки

Забезпечення безпеки об'єктів є одним з найважливіших аспектів у сучасному світі. Системи безпеки використовуються для захисту приватної власності, офісних приміщень, комерційних будівель, транспортних засобів та інших цінних майнових комплексів від несанкціонованого доступу, крадіжок, вандалізму та інших загроз.

Традиційні системи безпеки часто включають провідні сигналізаційні системи, відеоспостереження, контроль доступу та датчики руху. Проте, такі системи можуть бути дорогими у встановленні та підтримці, а також можуть бути обмеженими в своїй функціональності та можливостях розширення [1].

У зв'язку з цим, виникає потреба у більш сучасних та гнучких системах безпеки. Сучасні системи безпеки часто поєднують у собі різні компоненти: датчики, камери відеоспостереження, аудіосистеми, системи автоматичного сповіщення та системи контролю доступу. Вони забезпечують комплексний підхід до безпеки, забезпечуючи не тільки виявлення порушень, але й можливість вчасної реакції та взаємодії з операторами або власниками об'єкту.

Важливою деталлю є інтеграції систем безпеки з мережами зв'язку, що дозволяє віддалено керувати системою та отримувати сповіщення про події через мобільний зв'язок або Інтернет. Це відкриває нові можливості для моніторингу та управління системою безпеки навіть на відстані.

У даній роботі буде розроблена GSM сигналізація з використанням NodeMcu. Цей підхід поєднує в собі переваги бездротового зв'язку GSM, простоту управління та налаштування за допомогою NodeMcu. Це дозволить створити ефективну та доступну систему безпеки з можливістю дистанційного

керування та моніторингу.

### **1.1.2 Складові системи безпеки**

Система безпеки складається з різноманітних компонентів, які спільно працюють для забезпечення захисту об'єкту. Основні складові системи безпеки включають [1]:

- датчики безпеки використовуються для виявлення різних видів небезпек: відкриття дверей або вікон, дим, вогонь тощо; датчики можуть бути розташовані у всіх важливих точках об'єкту та передавати сигнали про виявлені небезпеки до центральної системи керування;
- центральна система керування – це мозок системи, який отримує інформацію від датчиків та приймає рішення про реалізацію необхідних заходів безпеки;
- сигналізація та сповіщення – ця складова системи безпеки включає в себе різні звукові, візуальні або вібраційні сигнали, які активуються в разі виявлення небезпеки. Вони можуть служити для сповіщення власника, охоронних служб або навколишніх людей про потенційну небезпеку.

### **1.1.3 Принцип роботи системи безпеки**

Система безпеки працює на основі певних принципів, які забезпечують її ефективність. Далі розглянемо ці принципи більш детально [1].

Виявлення – датчики системи безпеки постійно моніторять стан об'єкту і виявляють можливі небезпеки або порушення.

Аналіз – інформація, отримана від датчиків, проходить через центральну систему керування, де вона аналізується для визначення ступеня небезпеки та необхідних заходів.

Реагування – залежно від виявленої небезпеки, система сигналізації активує заходи безпеки, такі як сповіщення, включення сирени або автоматичне повідомлення служб безпеки.

Керування – власник об'єкту або відповідальні особи можуть керувати системою безпеки, налаштовувати параметри, отримувати звіти та відстежувати події в реальному часі.

#### **1.1.4 Типи датчиків безпеки**

Датчики безпеки використовуються для виявлення різних видів небезпек.

Поширені типи датчиків безпеки включають [2]:

- датчики руху вони фіксують рух у заданій зоні та використовуються для сповіщення або активації інших заходів безпеки;
- датчики відкриття вони виявляють відкриття дверей, вікон або інших доступних входів та вмикають сигналізацію;
- датчики диму та вогню вони виявляють наявність диму або вогню та негайно сповіщають про пожежу;
- датчики витоку газу вони реагують на наявність небезпечних газів і активують сповіщення або вмикають газові лінії;
- датчики протікання води вони виявляють протікання води та запобігають пошкодженню від затоплення.

#### **1.1.5 Центральна система керування**

Центральна система керування є серцевиною системи безпеки. Вона виконує важливі функції з отримання, аналізу та обробки інформації від датчиків безпеки, а також приймає рішення щодо подальших дій. Ця система відповідає за централізоване керування всіма компонентами системи безпеки та

забезпечує їх взаємодію.

Центральна система керування може включати контрольний пульт або інтерфейс, який дозволяє власнику об'єкту керувати системою безпеки. За допомогою контрольного пульта можна активувати або деактивувати систему, налаштовувати параметри роботи датчиків, отримувати інформацію про стан безпеки об'єкту. Це дає власнику повний контроль над системою та зручність у використанні [2].

Крім того, центральна система керування може мати можливість підключення до мережі зв'язку GSM, для передачі сповіщень або отримання дистанційних команд. Це дозволяє власнику отримувати сповіщення про події на свій мобільний телефон або інший пристрій.

## **1.2 Бездротовий зв'язок та технологія GSM**

### **1.2.1 Основи бездротового зв'язку**

Бездротовий зв'язок є ключовою технологією, що забезпечує передачу даних без використання проводів. Він використовує радіохвилі або інфрачервоне випромінювання для передачі інформації між пристроями. Бездротовий зв'язок дозволяє створювати мережі, підключати пристрої та обмінюватися даними без необхідності фізичного підключення.

Одним з популярних стандартів бездротового зв'язку є Global System for Mobile Communications (GSM). GSM є стандартом мобільного зв'язку, що використовується в більшості країн світу. Він дозволяє передавати голосові повідомлення, текстові повідомлення та дані через мобільні мережі [3].

Однією з ключових переваг GSM є широке покриття мережі, що дозволяє здійснювати зв'язок практично в будь-якому місці з належним сигналом. Крім того, GSM використовує SIM-карти для ідентифікації пристроїв. Це дозволяє керувати доступом до мережі та забезпечує безпеку передачі даних [3].

Використання GSM технології дозволяє забезпечити надійний зв'язок з системою сигналізації незалежно від місцезнаходження. Це дозволяє отримувати сповіщення про події, контролювати стан системи та надсилати сповіщення о тривозі через мобільну мережу. Застосування GSM забезпечує можливість віддаленого керування та моніторингу системи без необхідності фізичної присутності на об'єкті.

У розробці сигналізації використання GSM технології надасть широкі можливості для створення функціональної та ефективної системи безпеки.

### **1.2.2 GSM технологія та її особливості**

GSM (Global System for Mobile Communications) є одним з найпоширеніших стандартів мобільного зв'язку, який використовується у багатьох країнах світу. GSM технологія забезпечує передачу голосових даних, текстових повідомлень та даних через мобільну мережу [3].

Розглянемо основні особливості GSM технології більш детально [4].

Використання SIM-карт дозволяє ідентифікувати та автентифікувати користувачів у мережі. Кожен абонент має свою унікальну SIM-карту, яка забезпечує доступ до мережі і зберігає інформацію про абонента.

GSM мережа складається з базових станцій, які приймають та передають сигнали від мобільних пристроїв. Базові станції покривають певну територію, утворюючи мережу.

Захист інформації забезпечується шляхом шифрування передачі даних між мобільним пристроєм і базовою станцією. Це забезпечує конфіденційність даних та запобігає несанкціонованому доступу.

### **1.2.3 Застосування GSM в системах безпеки**

GSM технологія має широкі застосування в системах безпеки. Вона може бути використана для передачі сигналів тривоги, статусу безпеки та керування системою дистанційно. Далі розглянемо, де найчастіше зустрічається технологія GSM в системах безпеки [1].

GSM сигналізація – можливість передавати сигнали тривоги через GSM мережу, що дозволяє швидко та надійно повідомляти про вторгнення, пожежу або інші небезпечні ситуації. Це дозволяє власнику об'єкту або службі охорони отримувати сповіщення в реальному часі та приймати необхідні заходи.

Відеоспостереження – GSM технологія може бути використана для передачі відео з відеокамер безпеки на мобільні пристрої. Це дозволяє власнику здійснювати віддалений моніторинг об'єкту та вчасно реагувати на події.

Дистанційне керування – за допомогою GSM технології власники можуть керувати системою безпеки з віддаленої локації. Це включає включення або вимикання системи, налаштування параметрів, активацію або деактивацію датчиків та отримання статусних повідомлень.

Аналіз і розуміння бездротового зв'язку та особливостей GSM технології дозволяють визначити, як ці технології можуть бути використані в розробці сигналізації.

## **1.3 Мікроконтролер NodeMcu**

### **1.3.1 Огляд мікроконтролера**

Мікроконтролер NodeMcu є компактним пристроєм, побудованим на базі мікросхеми ESP8266 (див. рис. 1.1). Він поєднує в собі можливості мікроконтролера та вбудованого модуля Wi-Fi, що робить його ідеальним вибором для розробки проєктів, включаючи системи безпеки. NodeMcu став

популярним у сфері розробки IoT (Internet of Things) завдяки своїй простоті використання та розширеній функціональності [6].

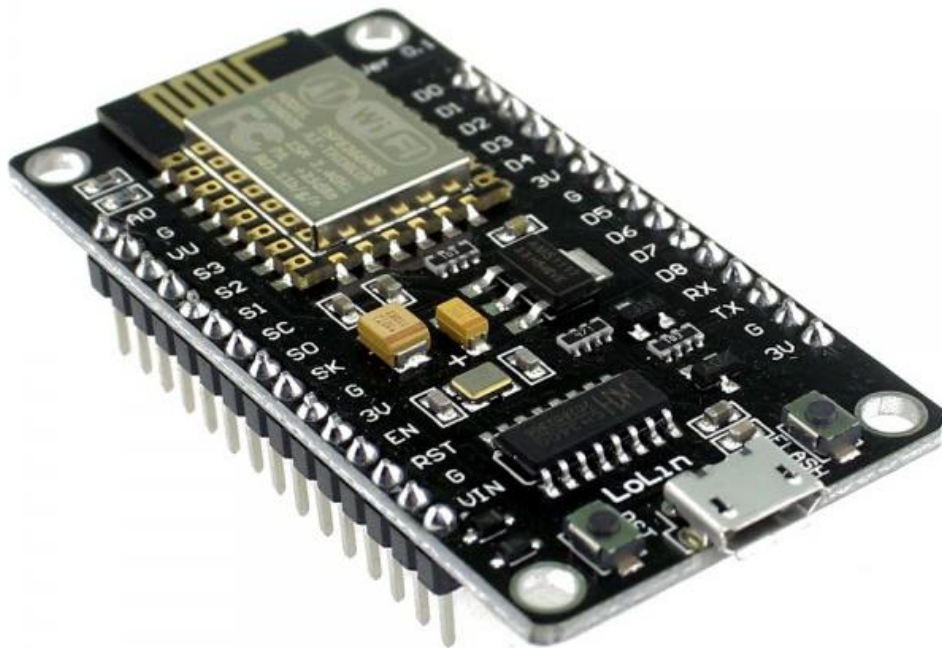


Рисунок 1.1 – Мікроконтролер NodeMcu

В NodeMcu є вбудований модуль Wi-Fi для підключення до бездротової мережі для взаємодії з іншими пристроями через Інтернет. Це дозволяє віддалено керувати системою безпеки, моніторити стан об'єктів та отримувати повідомлення про події з будь-якого місця з доступом до Інтернету.

NodeMcu має роз'єми для підключення додаткових сенсорів, реле та інших пристроїв, що дозволяє розширити функціональні можливості системи безпеки. Завдяки вбудованій пам'яті та можливості виконання програм, NodeMcu може функціонувати як незалежна система, без необхідності підключення до комп'ютера або сервера [6].

Використання NodeMcu у розробці сигналізації дозволяє простіше взаємодіяти з GSM модулем, встановлювати з'єднання з мережею, відправляти та отримувати SMS-повідомлення. NodeMcu забезпечує гнучкість у програмуванні та налаштуванні системи, що дозволяє адаптувати її під конкретні потреби безпеки об'єктів.



У розробці сигналізації використання цього мікроконтролера дозволяє створити компактну та зручну систему, забезпечуючи бездротовий зв'язок та програмовану логіку функціонування.

### **1.3.2 Архітектура та характеристики**

Мікроконтролер NodeMcu володіє потужною архітектурою та характеристиками, що роблять його привабливим для розробки систем безпеки. Далі розглянемо основні характеристики мікроконтролера NodeMcu детальніше.

Процесор мікроконтролера NodeMcu має 32-бітний процесор, який працює на тактовій частоті 80 МГц. Це забезпечує швидку та ефективну обробку даних, що є важливим аспектом для систем безпеки [6].

Пам'ять мікроконтролера має 4 МБ флеш-пам'яті, яка використовується для зберігання програмного коду та даних. Ця пам'ять дозволяє зберігати необхідні конфігураційні файли, бібліотеки та інші дані, необхідні для роботи системи безпеки [6].

Крім внутрішньої флеш-пам'яті, NodeMcu підтримує зовнішню пам'ять типу SPI Flash. Це дозволяє розширити обсяг доступної пам'яті для зберігання додаткових даних або програмного коду [5].

Ці особливості мікроконтролера NodeMcu роблять його потужним інструментом для розробки систем безпеки. Висока швидкість обробки даних та достатня пам'ять дозволяють виконувати складні алгоритми та забезпечувати надійну роботу системи сигналізації.

## 1.4 Огляд інструменту розробки

### 1.4.1 Опис Arduino IDE

Arduino IDE (Integrated Development Environment) – це безкоштовне інтегроване середовище розробки, яке надає зручність та ефективність для програмування мікроконтролерів, зокрема NodeMcu. Воно є одним з найпоширеніших та найбільш доступних інструментів для розробки проєктів з використанням Arduino-сумісних платформ [7].

Arduino IDE пропонує простий та зрозумілий інтерфейс користувача, що робить його популярним серед початківців у світі мікроконтролерного програмування. Основна філософія Arduino IDE полягає у спрощенні процесу програмування, щоб дозволити інженерам та студентам швидко створювати та втілювати свої ідеї.

Структура та основні компоненти Arduino IDE (див. рис. 1.2) [7]:

- Arduino IDE надає зручне вікно редактору коду для написання, редагування та перегляду коду; воно включає основні функції редагування тексту, такі як виділення синтаксичних елементів, автодоповнення та перевірку синтаксису;
- панель інструментів містить основні кнопки та команди для компіляції, завантаження коду на мікроконтролер, відкриття/збереження проєктів та налаштування параметрів;
- монітор порту дозволяє переглядати вивідні дані з мікроконтролера, наприклад, повідомлення про помилки, дані з датчика; він також дозволяє надсилати команди до мікроконтролера для інтерактивної взаємодії;
- має вбудований менеджер платформ та бібліотек, який дозволяє легко встановлювати та оновлювати підтримувані платформи та бібліотеки для розробки – це спрощує процес налаштування середовища для конкретного мікроконтролера;
- Arduino IDE надає засоби для здійснення серійної комунікації з

мікроконтролером – це дозволяє передавати та отримувати дані між комп'ютером та мікроконтролером, що відкриває можливості для зовнішнього керування та моніторингу;

- заготовки (скетчі) – це невеликі програми, які виконуються на мікроконтролері: вони містять основну структуру коду, включаючи функцію `setup()` (для налаштування) та функцію `loop()` (для повторюваних операцій).



Рисунок 1.2 – Структура Arduino IDE

## 1.4.2 Встановлення середі розробки Arduino IDE

Для розробки на мікроконтролерах необхідно встановити Arduino IDE на комп'ютер. Встановлення Arduino IDE включає наступні кроки [8]:

- 1) завантаження Arduino IDE – перш за все, необхідно завантажити Arduino IDE з офіційного веб-сайту Arduino (існують версії для різних операційних систем, таких як Windows, macOS та Linux, виберіть версію, яка відповідає вашій операційній системі);
- 2) встановлення Arduino IDE – після завантаження виконайте файл інсталяції Arduino IDE (слідуйте інструкціям у майстері встановлення

- для обрання місця встановлення та інших налаштувань, процес встановлення зазвичай є простим і не вимагає складних налаштувань);
- 3) підключення до Arduino – після встановлення підключіть ваш мікроконтролер Arduino до комп'ютера за допомогою USB-кабелю (впевніться, що мікроконтролер правильно підключений і розпізнаний системою);
  - 4) налаштування платформи – перед початком роботи з Arduino IDE необхідно налаштувати платформу для вашого мікроконтролера (у верхній частині вікна Arduino IDE виберіть потрібну платформу зі списку доступних платформ – це дозволить IDE правильно інтерпретувати код та встановити необхідні налаштування для конкретного мікроконтролера);
  - 5) перевірка роботи – після успішного встановлення та налаштування Arduino IDE, ви можете виконати простий тест, щоб переконатися, що воно працює належним чином (спробуйте запустити один з вбудованих прикладів, які поставляються з Arduino IDE – це допоможе перевірити з'єднання з мікроконтролером і правильність налаштувань).

### **1.4.3 Налаштування середовища розробки для плати ESP8266**

За замовчуванням в Arduino IDE немає плати ESP8266. Тому потрібно додати додаткові пакети в середу розробки.

Для цього відкриваємо Arduino IDE і додаємо можливість роботи з платами esp8266 [5]:

- 1) переходимо до розділу «Preference» у меню;
- 2) у поле «Additional boards manager URLs» вставляємо рядок «[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)» і натискаємо кнопку «OK» (див. рис. 1.3);

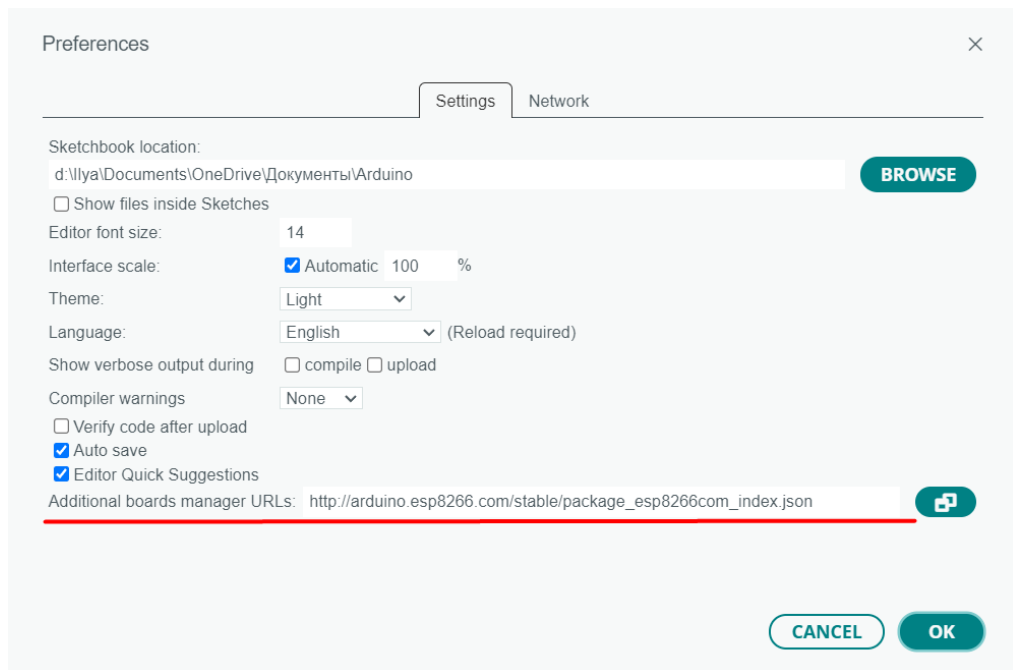


Рисунок 1.3 – Налаштування для роботи з платою ESP8266

- 3) відкриваємо плату менеджера в меню «Tools > Board > Boards Manager» (див. рис. 1.4);

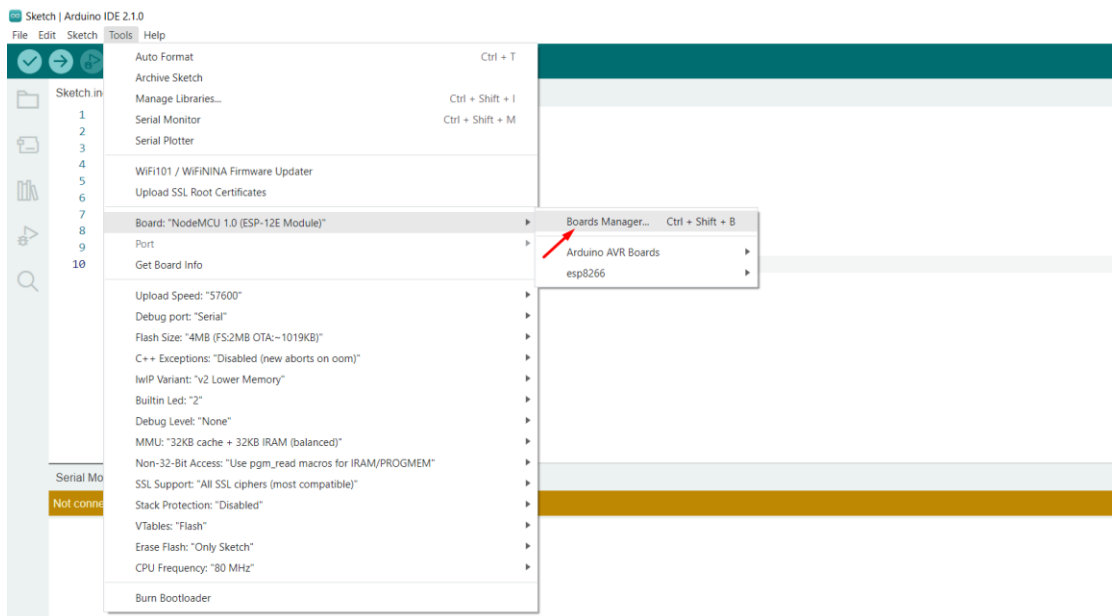


Рисунок 1.4 – Інструкція з налаштування

- 4) у вікні, що з'явилось, в пошуку вводимо «esp8266» і встановлюємо відповідний пакет (див. рис. 1.5);

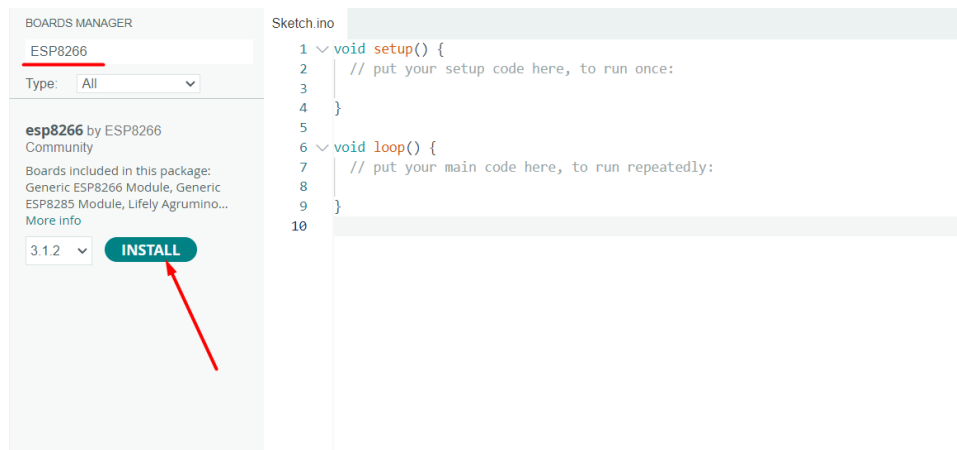


Рисунок 1.5 – Встановлення пакету

- 5) дочекайтеся завершення встановлення збірки, після успішного встановлення пакету, у списку, навпроти її назви «esp8266 by ESP8266» з'явиться фраза «INSTALLED».

На цьому налаштування Arduino IDE завершено! Тепер у списку плат є розділ ESP8266 Boards.

## **2 ПРОЄКТУВАННЯ СИСТЕМИ GSM СИГНАЛІЗАЦІЇ**

### **2.1 Вимоги до системи сигналізації**

Вимоги до системи сигналізації включають надійність, ефективність, гнучкість, інтегрованість, простота використання, безпека даних та масштабованість [2].

Система повинна працювати бездоганно і забезпечувати стабільний захист об'єкту від потенційних загроз. Вона має забезпечувати швидке та точне виявлення загроз, реагувати та передавати інформацію про події в реальному часі.

Гнучкість системи полягає в її здатності адаптуватись до різних потреб та контексту. Інтегрованість забезпечує співпрацю з іншими системами безпеки, що сприяє комплексному підходу.

Простота використання дозволяє користувачу легко керувати системою без глибоких знань. Безпека даних забезпечує конфіденційність та цілісність інформації. Масштабованість передбачає здатність системи розширюватись та адаптуватись до змінних потреб та обсягу території, що охороняється. Ці вимоги формуються з урахуванням потреб замовника та специфікацій проєкту, забезпечуючи найвищий рівень безпеки.

#### **2.1.1 Визначення функціональних вимог**

Функціональні вимоги до системи сигналізації відіграють важливу роль у розробці та налагодженні ефективної системи безпеки. Ці вимоги визначають очікувані функціональні можливості та характеристики системи, що забезпечують надійний захист об'єкту. Вони описують, як система має виявляти загрози, сповіщати про події, виконувати автоматичні дії та

забезпечувати взаємодію з користувачем.

В даній роботі були визначені наступні функціональні вимоги до системи сигналізації:

- виявлення руху: система повинна бути здатна виявляти рух у встановленій зоні та спрацьовувати, коли виявлено неправомірний рух, для цього використовуються рухові датчики, які реагують на зміни в інфрачервоному або радіочастотному спектрі;
- моніторинг стану системи: система сигналізації постійно моніторить свій стан для забезпечення правильної роботи: перевірку статусу, перевірку доступності GSM-зв'язку, стану батареї, працездатності логіки системи та інших важливих параметрів, якщо будь-який з цих параметрів виявляє проблему, система повинна вжити відповідних заходів;
- сповіщення про події: після спрацювання сигналізації, система повинна негайно сповістити про виникнення події – це може бути здійснено за допомогою надсилання повідомлень користувачеві через SMS-повідомлення;
- дистанційне керування: користувач повинен мати можливість дистанційно керувати системою: включення/виключення, налаштування параметрів та отримання статусу безпеки через мобільний додаток.

Ці функціональні вимоги визначають основні можливості, які система сигналізації повинна мати з метою забезпечення безпеки об'єкту та оперативного реагування на потенційні загрози.

### **2.1.2 Вимоги до безпеки та надійності**

Вимоги до безпеки та надійності системи сигналізації є ключовими аспектами для забезпечення ефективності та надійності її роботи. Несанкціонований доступ до системи має бути ефективно унеможливлений за допомогою механізмів захисту та використанням стандартів безпеки.



Вимоги до безпеки включають такі аспекти [2]:

- рухові датчики повинні працювати надійно і точно виявляти рух у встановленій зоні безпеки, вони мають бути стійкими до нежаданих спрацювань, спричинених шумами або коливаннями навколишнього середовища;
- система повинна мати механізми захисту, щоб запобігти несанкціонованому доступу до системи та її функцій, це може включати використання паролів або інших методів автентифікації для доступу до системи, а також захист комунікаційних каналів для запобігання перехопленню або зламу інформації;
- вона повинна мати надійні засоби комунікації, які забезпечують швидку і безперебійну передачу повідомлень про події, це може включати використання GSM-зв'язку, Інтернет-протоколів або інших засобів зв'язку з високою надійністю;
- сигналізація повинна мати резервне електроживлення або підтримку батарейного живлення, щоб забезпечити безперебійну роботу в разі відключення основного джерела електроенергії, що дозволить системі продовжувати працювати та здійснювати моніторинг навіть під час перебоїв електропостачання.

Забезпечення безпеки та надійності системи сигналізації є важливими аспектами, які допомагають гарантувати захист майна та життя, а також забезпечують ефективну та надійну роботу системи у будь-яких умовах.

## **2.2 Вибір апаратних компонентів**

У процесі розробки системи сигналізації було проведено вибір апаратних компонентів, які відповідають вимогам і забезпечують необхідну функціональність системи. Основні компоненти, які були вибрані, включають мікроконтролер NodeMcu, GSM-модуль та сенсори для виявлення порушень

безпеки.

### 2.2.1 Вибір мікроконтролера NodeMcu

При виборі мікроконтролера був обраний NodeMcu, який є популярним і потужним мікроконтролером на основі чіпу ESP8266 з підтримкою Wi-Fi (див. рис. 2.1). Цей вибір був здійснений з урахуванням різних факторів, що забезпечують оптимальну роботу системи.

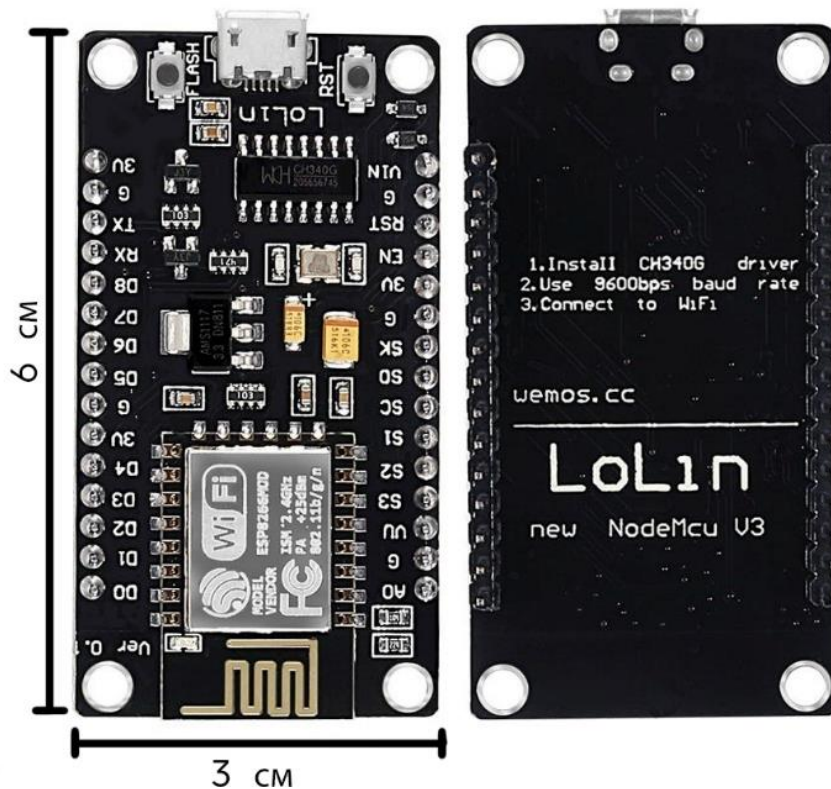


Рисунок 2.1 – Модуль NodeMcu V3 ESP8266

NodeMcu забезпечує зручний програмний інтерфейс для розробки на мові Lua або Arduino IDE, що спрощує процес програмування і розробки функціональності системи сигналізації. Це робить його доступним для широкого кола розробників та дозволяє швидше створювати програмне забезпечення [6].

Мікроконтролер має вбудовану пам'ять для збереження програми, що



Таблиця 2.1 – Конфігурація контактів плати NodeMcu [6]

PIN Category	Name	Description
Power	Micro-USB, VU, 3.3V, Vin	Micro-USB – можна жити через порт USB Vin – вивід для підключення зовнішнього джерела живлення 5V. VU – вивід, на який заведено напругу 5V з USB-роз'єму. 3.3V – контакт, на який подається вихідна напруга. Може бути використаний для живлення датчиків, що підключаються до плати.
Control pins	RST, EN	Пін і кнопка скидання мікроконтролер
Analog Pin	A0	Використовується для вимірювання аналогової напруги в діапазоні 0-3,3 В
GPIO Pins	GPIO1 to GPIO16	16 контактів вводу-виводу загального призначення на своїй платі
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU має два інтерфейси UART: UART0 (RXD0 і TXD0) і UART1 (RXD1 і TXD1).

### 2.2.2 Вибір GSM модуля для забезпечення зв'язку

Для забезпечення зв'язку системи сигналізації був обраний GSM-модуль SIM800L (див. рис. 2.3). Цей модуль володіє необхідними характеристиками для надійного та ефективного забезпечення зв'язку між системою та користувачем.

Основною перевагою модуля SIM800L є підтримка GSM-мереж, що дозволяє передавати дані через мобільний зв'язок. Він підтримує стандарти GSM, GPRS, а також може працювати у різних діапазонах частот. Це

забезпечує широкий радіус дії системи сигналізації [5].

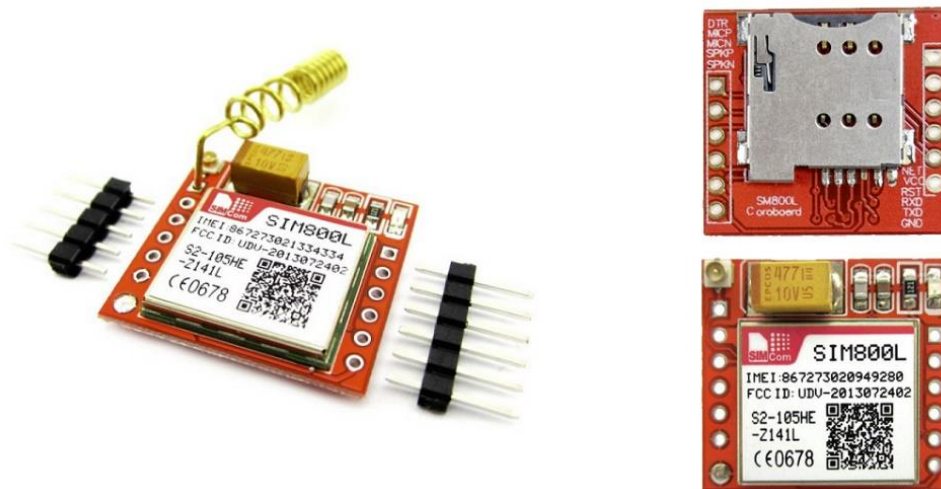


Рисунок 2.3 – Модуль GSM SIM800L

Модуль SIM800L має компактні розміри і низьку споживану потужність (див. табл. 2.2), що робить його ідеальним вибором для використання в системах, де обмежені ресурси. Це дозволяє легко впроваджувати модуль у систему сигналізації без значних зусиль [9].

Таблиця 2.2 – Основні характеристики модуля SIM800L

Особливість	Інформація
Напруга живлення	3.4V ~ 4.4V
Робоча температура	-30°C ~ +75°C
Швидкість UART:	1200-115200 бод
Режим мережі	2G
Розміри	5 x 25 мм
SIM	Підтримує сімкарти з живленням 3В та 1.8В

Зокрема, щодо живлення, важливо зазначити, що живлення модуля SIM800L та плати NodeMcu, повинні бути забезпечені окремо. NodeMcu може працювати з напругою 3.3 Вольта. Вбудований регулятор напруги NodeMcu

забезпечує стабільну роботу мікроконтролера та інших компонентів при цій напрузі. Що стосується модуля SIM800L, для його живлення потрібна напруга в діапазоні 3.4 - 4.4 Вольта. Таким чином, NodeMcu не може безпосередньо жити модуль SIM800L, оскільки його напруга живлення становить 3.3 Вольта, що нижче мінімальної необхідної напруги для SIM800L. Тому для забезпечення живлення GSM-модуля було вирішено використати літій-іонний акумулятор.

Літій-іонні акумулятори є поширеним джерелом живлення для портативного електронного обладнання завдяки своїй високій ємності, малій вазі та тривалому часу роботи (див. табл. 2.3) [13].

Таблиця 2.3 – Характеристики літій-іонного акумулятора.

Характеристика	Особливість
Енергетична місткість	110 - 200 Вт·год/кг
Напруга максимальна в елементі	4,18 - 4,20 В
Напруга мінімальна	2,5 - 2,75 В
Діапазон робочих температур	- 20 - + 60° С
Ресурс	500 - 1000 циклів заряду

SIM800L також підтримує передачу SMS-повідомлень, що є важливою функціональністю для системи сигналізації. Він може надсилати сповіщення користувачеві через SMS у разі спрацювання сигналізації або виникнення інших подій [9].

SIM800L має ряд пінів (див. рис. 2.4), які використовуються для забезпечення зв'язку з модулем та підключення до зовнішніх пристроїв.

Основні піни, які варто знати і які нам знадобляться в подальшій розробці системи сигналізації включають наступні [9]:

- VCC (використовується для живлення модуля SIM800L);
- GND (земляний пін, який повинен бути підключений до землі або негативного полюсу джерела живлення);

- TXD, RXD (використовуються для зв'язку з мікроконтролером. TXD (Transmit Data) передає дані від модуля, а RXD (Receive Data) отримує дані від зовнішнього пристрою);
- RST (відповідає за скидання модуля SIM800L, підключення цього піна до рівня HIGH (напруга 3.7 – 4.2 Вольт) протягом короткого періоду часу призведе до скидання модуля);
- SIM Card Slot (SIM800L має слот для встановлення SIM-карти, яка використовується для автентифікації в мережі оператора зв'язку).

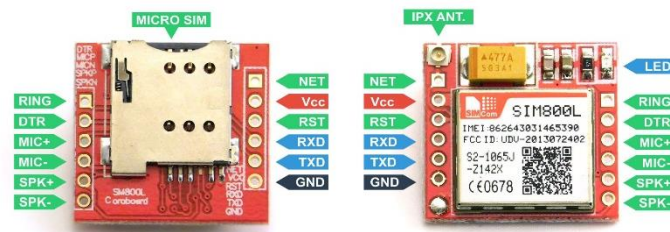


Рисунок 2.4 – Розпізнавання контактів модуля SIM800L

Вибір GSM-модуля SIM800L є обґрунтованим з точки зору забезпечення надійного, ефективного та універсального зв'язку для системи сигналізації. Його функціональність, компактність та підтримка різних стандартів роботи роблять його відмінним вибором для забезпечення зв'язку з користувачем.

### 2.2.3 Вибір сенсорів для виявлення порушень безпеки

У системі сигналізації було обрано декілька сенсорів для виявлення порушень безпеки, включаючи датчик руху HC-SR501, п'єзовипромінювач KY-012 та світлодіод.

Датчик руху HC-SR501 (див. рис. 2.5) є популярним і надійним компонентом для виявлення руху у зазначеній зоні. Він працює на основі інфрачервоного детектування руху та здатний виявляти рух об'єктів у встановленому діапазоні. Коли датчик реєструє рух, він генерує сигнал, що

вмикає сигналізацію та запускає відповідні заходи захисту. Датчик руху є ефективним і простим у використанні для виявлення потенційно небезпечних ситуацій [10].



Рисунок 2.5 – Датчик руху HC-SR501

У датчика наявне регулювання чутливості за допомогою якого можна визначати рух об'єктів на відстані від трьох до семи метрів (див. табл. 2.4). Датчик оснащений трьома коннекторами на задній частині. Один вихід йде до землі (GND), другий – забезпечує видачу необхідного сигналу з датчику (+Power), а третій є цифровим виходом, з якого знімаються дані (див. рис. 2.6) [10].

Таблиця 2.4 – Основні характеристики датчику руху HC-SR501

Особливість	Інформація
Напруга живлення	5V - 20V
Робоча температура	-15°C ~ +70°C
Розміри	32x24 мм
Кут огляду	менше 120 градусів, в межах 7 метрів





Рисунок 2.6 – Розпізнавання датчика руху

П'єзовипромінювач KY-012 (див. рис. 2.7) має компактний розмір і простий у використанні.

П'єзовипромінювач є аудіо-візуальним пристроєм, який використовується для негайного сповіщення про спрацювання системи сигналізації. Коли сигналізація спрацьовує, модуль генерує гучний звуковий сигнал, що привертає увагу інших людей. Це допомагає забезпечити своєчасне реагування на події та запобігти можливим небезпекам [11].



Рисунок 2.7 – П'єзовипромінювач KY-012

Світлодіод (англ. LED – light-emitting diode) – це напівпровідниковий пристрій, що при пропусканні через нього електричного струму і певної хімічної реакції випромінює світло. Світлодіод є світловим індикатором, який також використовується для візуального сповіщення про стан системи

сигналізації. Він може випромінювати світло різних кольорів, що вказує на різні стани системи. Щоб забезпечити правильну роботу світлодіода і запобігти пошкодженню, до нього потрібно підключити резистор. Зазвичай для світлодіода використовується резистор зі значенням 220 Ом. Резистор обмежує потік струму через світлодіод, що допомагає зберегти його цілісність і продовжити його робочий час [12].

Вибір цих сенсорів заснований на їхній надійності, доступності, вартості та сумісності з мікроконтролером NodeMcu. Вони утворюють комплексну систему виявлення порушень безпеки.

## **2.3 Розробка архітектури системи**

### **2.3.1 Опис загальної структури системи**

Опис загальної структури системи включає в себе детальний опис компонентів, їх функцій та взаємодії. Далі розглянемо основні компоненти системи сигналізації детальніше.

Користувач – це особа або група осіб, які мають доступ до системи сигналізації. Користувач може взаємодіяти з системою через мобільний додаток. Вони мають можливість активувати або деактивувати сигналізацію, контролювати стан системи, отримувати сповіщення про події та відправляти команди до системи [2].

Датчики безпеки – це фізичні пристрої, які використовуються для виявлення можливих порушень безпеки. Вони можуть включати датчики руху, датчики відкриття дверей або вікон, датчики витоку газу або диму та інші. Вони реагують на зміни в оточуючому середовищі та передають сигнали про події до центральної системи [2].

Центральна система – це основний модуль системи сигналізації, який приймає, обробляє та керує інформацією від датчиків безпеки та користувачів.

Вона може бути фізичним пристроєм або програмним компонентом. Центральна система відповідає за аналіз сигналів від датчиків безпеки, визначення стану системи (активована/деактивована), прийняття рішень щодо спрацювання сигналізації та надсилання сповіщень користувачам [2].

Сповіщення – це механізм передачі сповіщень від системи сигналізації до користувачів. Сповіщення можуть бути надіслані через канали зв'язку, такі як SMS-повідомлення. Вони містять інформацію про стан системи, виявлені події та інші важливі повідомлення для користувача.

Ця загальна структура системи сигналізації дозволяє користувачам ефективно контролювати та забезпечувати безпеку своїх приміщень, отримувати оперативну інформацію про події та при необхідності приймати відповідні заходи.

Представимо цю структуру системи у вигляді діаграми використання (див. рис. 2.8) :

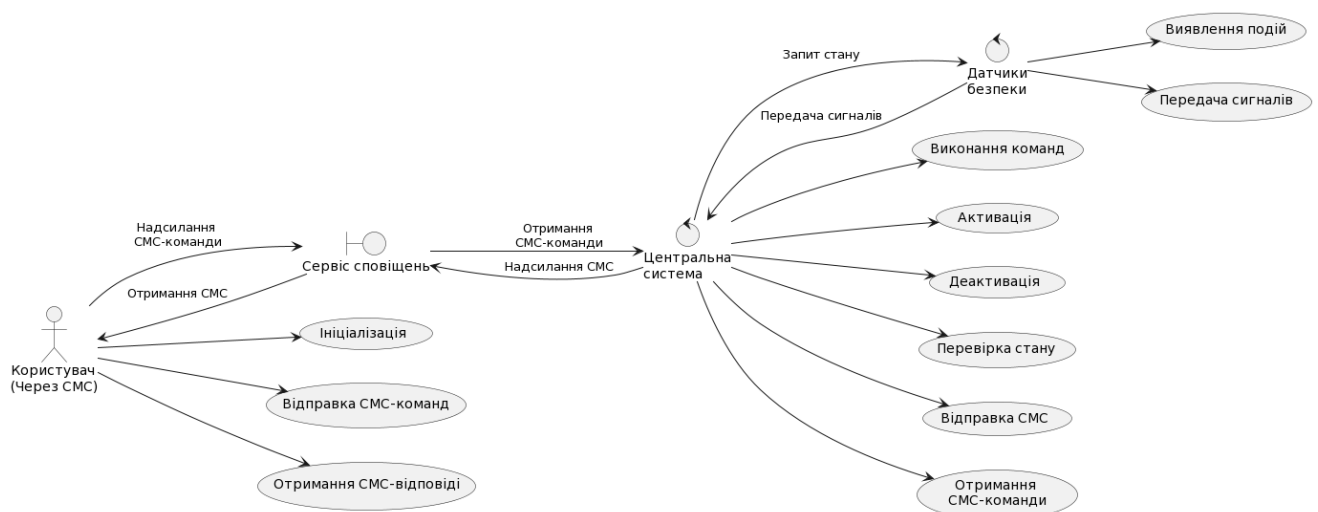


Рисунок 2.8 – Діаграма використання для системи сигналізації

### 2.3.2 Опис взаємодії між компонентами системи

В цьому пункті опишемо взаємодію між компонентами системи.

Мікроконтролер виконує центральну роль у системі сигналізації. Він

координує роботу всіх компонентів і приймає рішення на основі отриманої інформації. NodeMcu забезпечує зчитування даних від датчиків, керування активацією та деактивацією системи, відправку повідомлень і контроль роботи інших компонентів.

GSM-модуль – використовується для забезпечення зв'язку системи сигналізації з користувачем через мобільний зв'язок. Він отримує команди від мікроконтролера NodeMcu і надсилає SMS-повідомлення користувачеві і навпаки, він може отримувати SMS-команди від користувача і передавати їх до NodeMcu для виконання відповідних дій.

Датчик руху – використовується для виявлення руху в приміщенні. Він відправляє сигнал до мікроконтролера, коли рух був виявлений. Це дозволяє системі сигналізації реагувати на незаконне проникнення у приміщення.

П'єзовипромінювач – використовується для генерації звукового сигналу в разі спрацювання системи сигналізації. Коли мікроконтролер NodeMcu отримує сигнал про активність, він вмикає п'єзовипромінювач для відтворення звукового сигналу, що привертає увагу оточуючих і нагадує про виникнення події.

Світлодіод використовується для візуального сигналізування стану системи сигналізації. Він може бути ввімкнений або вимкнений в залежності від стану системи. Світлодіод увімкнений, коли система активована, і вимкнений, коли система деактивована.

Взаємодія між компонентами системи відбувається наступним чином:

- 1) мікроконтролер NodeMcu зчитує дані від датчика руху, перевіряє їх на наявність руху і якщо рух був виявлений, то активує систему сигналізації;
- 2) після активації системи, NodeMcu відправляє команду до GSM-модуля SIM800L для надсилання SMS-повідомлення користувачеві про спрацювання сигналізації;
- 3) в разі спрацювання системи NodeMcu вмикає п'єзовипромінювач для генерації звукового сигналу і вмикає світлодіод для візуального

сигналізування;

- 4) користувач може відправляти SMS-команди на GSM-модуль SIM800L для виконання певних дій, наприклад, активації або деактивації системи. SIM800L передає отриману команду до NodeMcu для її виконання;
- 5) в разі отримання важливих повідомлень від GSM-модуля SIM800L, NodeMcu може надіслати SMS-повідомлення користувачеві, що містить інформацію про стан системи або події.

Таким чином, всі компоненти системи взаємодіють між собою для забезпечення надійного та ефективного функціонування системи сигналізації. Команди, стан і повідомлення передаються між мікроконтролером та GSM-модулем, а датчик руху, п'єзовипромінювач і світлодіод виконують відповідні дії на основі команд та стану системи.

## 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ СИГНАЛІЗАЦІЇ

### 3.1 Розробка схеми підключення компонентів

Під час розробки схеми підключення компонентів для системи сигналізації важливо врахувати правильність з'єднань та живлення кожного компонента.

Після визначення компонентів системи сигналізації та теоретичної інформації у попередніх підрозділах, можна приступати до створення схеми підключення елементів системи сигналізації, яка складається з мікроконтролера NodeMcu, датчика руху HC-SR501, GSM модуля SIM800L, літій іонного акумулятора, п'єзовипромінювача KY-012, світлодіод та резистора на 220 Ом. Усі елементи схеми підключаються між собою за допомогою проводів.

Детальний опис підключення компонентів (див. рис. 3.1):

- а) мікроконтролер NodeMcu і GSM модуль SIM800L:
  - 1) пін D7 на NodeMcu підключається до піна RX (прийом даних) на SIM800L модулі;
  - 2) пін D8 на NodeMcu підключається до піна TX (передача даних) на SIM800L модулі;
  - 3) земля (GND) на NodeMcu підключається до землі на SIM800L модулі;
- б) NodeMcu і датчик руху HC-SR501:
  - 1) пін OUT (вихідний сигнал) на датчику руху HC-SR501 підключається до піна D3 на NodeMcu;
  - 2) земля (GND) на датчику руху HC-SR501 підключається до землі на NodeMcu;
  - 3) живлення (+Power) на датчику руху HC-SR501 підключається до піна VU на NodeMcu;
- в) NodeMcu і п'єзовипромінювач KY-012:

- 1) пін SIG (сигнал) на п'езовипромінювачі KY-012 підключається до пина D1 на NodeMcu;
  - 2) земля (GND) на п'езовипромінювачі KY-012 підключається до землі на NodeMcu;
  - 3) живлення (VCC) на п'езовипромінювачі KY-012 підключається до живлення 3.3V на NodeMcu;
- г) світлодіод:
- 1) анод (довга ніжка) світлодіода підключається до вихідного пина D0 на NodeMcu через резистор 220 Ом для обмеження струму;
  - 2) катод (коротка ніжка) світлодіода підключається до землі на NodeMcu;
- д) літій-іонний акумулятор:
- 1) позитивний (+) полюс акумулятора підключається до живлення (VCC) на SIM800L модулі;
  - 2) негативний (-) полюс акумулятора підключається до землі на SIM800L модулі.

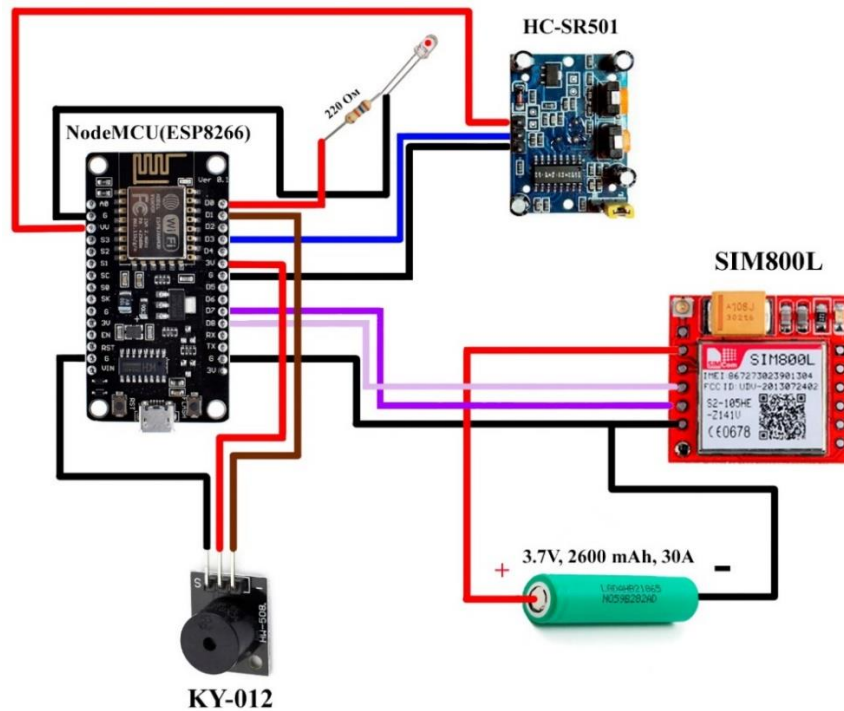


Рисунок 3.1 – Схема підключення компонентів системи сигналізації

## 3.2 Реалізація програмного забезпечення

Реалізація програмного забезпечення для системи сигналізації була виконана на мові програмування C++ з використанням Arduino IDE.

Arduino IDE – це середовище розробки, спеціально призначене для програмування мікроконтролерів Arduino та сумісних платформ, включаючи NodeMcu, який базується на мікроконтролері ESP8266 [7].

### 3.2.1 Конфігурація мікроконтролера NodeMcu

Для належної роботи системи сигналізації з використанням мікроконтролера NodeMcu, необхідно здійснити його налагодження та налаштування. Основні кроки конфігурації NodeMcu розглянемо нижче.

Для початку підключаємо бібліотеку «SoftwareSerial» (див. рис. 3.2), яка дозволяє створити віртуальний порт для комунікації з GSM модулем через певні піни мікроконтролера NodeMcu [14].

```
#include <SoftwareSerial.h>  
  
SoftwareSerial gsmModule(D7, D8); // RX=D7, TX=D8 pins for GSM module
```

Рисунок 3.2 – Підключення бібліотеки

Наступним кроком визначаємо піни – задаються змінні для пінів мікроконтролера NodeMcu, які використовуються для підключення компонентів системи сигналізації (датчик руху, світлодіод, п'єзовипромінювач). Також додаємо номери телефонів адміністратора (див. рис. 3.3).



```

const int motionPin = D3; // Motion sensor pin, connected to D3
const int buzzerPin = D1; // Buzzer pin, connected to D1
const int ledPin = D0; // LED pin, connected to D0

String adminNumbers[] = {"+380123456789"}; // Record administrator phone numbers, you can add more than one

```

Рисунок 3.3 – Ініціалізація пінів

Після ініціалізації системи та оголошення глобальних змінних, система потребує налаштування певних параметрів, які мають бути виконані першочергово після появи живлення. Це відбувається у функції «Setup», вона викликається один раз під час запуску мікроконтролера NodeMcu і використовується для початкової конфігурації системи та налаштування необхідних параметрів перед початком роботи [5]. Першим кроком в цьому процесі є налаштування монітора Serial, який відповідає за виведення повідомлень про роботу системи. Другим кроком налаштовуємо режим роботи пінів (див. рис. 3.4).

```

pinMode(motionPin, INPUT); // Set motionPin as input
pinMode(ledPin, OUTPUT); // Set ledPin as output
pinMode(buzzerPin, OUTPUT); // Set buzzerPin as output

```

Рисунок 3.4 – Режим роботи пінів

Далі ініціалізуємо зв'язок з GSM модулем за допомогою «gsmModule.begin(9600)» Це дозволяє мікроконтролеру встановити зв'язок з GSM модулем зі швидкістю передачі даних 9600 бод. Після відправляємо команду до GSM модуля для його налаштування. gsmModule.println(«AT») відправляє команду «AT» до GSM модуля для перевірки зв'язку (див. рис. 3.5) [15]. Після відправки команди, перевіряється наявність даних в буфері вхідного потоку. Якщо дані доступні, вони виводяться в монітор Serial.

```

gsmModule.begin(9600); // Initialize the GSM module communication
delay(1000);

gsmModule.println("AT"); // Send AT command to the GSM module
delay(1000);
while (gsmModule.available()) {
  Serial.write(gsmModule.read()); // Print the response from the GSM module to Serial Monitor
}

```

Рисунок 3.5 – Ініціалізація та перевірка GSM модуля

Далі налаштуємо текстовий режим SMS за допомогою команди «AT» команди – «AT+CMGF=1». Це дозволяє використовувати текстовий режим для обміну SMS повідомленнями. Налаштування режиму сповіщення про SMS за допомогою команди «AT+CNMI=1,2,0,0,0» (див. рис. 3.6). Це встановлює режим сповіщення, при якому модуль GSM надсилатиме сповіщення про нові SMS повідомлення [15]. Також відправляємо СМС адміністратору о тому, що система ініціалізована та готова до роботи.

```

gsmModule.println("AT+CMGF=1"); // Set SMS text mode
delay(1000);
while (gsmModule.available()) {
  Serial.write(gsmModule.read());
}

gsmModule.println("AT+CNMI=1,2,0,0,0"); // Set up SMS notification mode
delay(1000);
while (gsmModule.available()) {
  Serial.write(gsmModule.read());
}

// Delete all SMS messages
gsmModule.println("AT+CMGD=1,4"); // Delete all messages in all storage locations
delay(1000);
while (gsmModule.available()) {
  Serial.write(gsmModule.read());
}

sendSMS("Alarm system initialized. Enter 'On' to arm the alarm, 'Off' to disarm, and 'Battery' to check battery status.");
// Send an SMS to notify the administrators about the initialization and provide usage instructions.
}

```

Рисунок 3.6 – Налаштування GSM модуля

Тепер необхідно задати основну логіку роботи системи, яка буде працювати в нескінченному циклі. У цій частині коду задаються параметри роботи програми (див. рис. 3.7).

```

void loop() {

    processSMSCommands(); // check for user input and process commands
    checkMotionSensor(); // check motion sensor and send SMS if motion detected

    // check battery level every 12 hours and send SMS if critically lowOff
    unsigned long currentTime = millis();
    if (currentTime - lastBatteryCheckTime >= checkInterval) {
        checkBattery();
        lastBatteryCheckTime = currentTime;
    }
}

```

Рисунок 3.7 – Основна функція

### 3.2.2 Реалізація функції датчика руху

Реалізація функції датчика руху відбувається в функції `checkMotionSensor()` (див. рис. 3.8). Основна мета цієї функції - перевіряти стан датчика руху та, у разі виявлення руху, вмикати сигналізацію і відправляти SMS повідомлення.

```

// Checks the motion sensor and triggers the alarm if motion is detected
void checkMotionSensor() {
    // Check if the alarm is armed and the motion sensor is triggered
    if (isArmed && !isMotionDetected && digitalRead(motionPin) == HIGH) {
        // Motion sensor detected motion
        isMotionDetected = true; // Set the motion sensor status to true
        digitalWrite(buzzerPin, LOW); // Turn off the buzzer
        digitalWrite(ledPin, HIGH); // Turn on the LED
        sendSMS("Alarm! Motion detected"); // Send an SMS notification for motion detection
    }
    // If motion is detected, blink the LED and the buzzer
    if (isMotionDetected) {
        digitalWrite(ledPin, !digitalRead(ledPin)); // Toggle the LED state
        delay(500);
        digitalWrite(buzzerPin, !digitalRead(buzzerPin)); // Toggle the buzzer state
        delay(100);
    }
}

```

Рисунок 3.8 – Функція датчика руху

### 3.2.3 Реалізація функції перевірки стану батареї

Функція стану батареї виконує перевірку рівня заряду батареї, щоб визначити її поточний стан (див. рис. 3.9). Реалізація цієї функції:

GSM модуль викликає команду «AT+CBC», для читання напруги. У відповідь надходить стан батареї у вигляді рядка, наприклад «+CBC: 0, 100, 4134». У цьому рядку міститься три параметри.

Перший параметр вказує на стан заряду батареї. Значення «0» означає, що батарея не заряджається, «1» – заряджається, «2» - зарядка закінчена.

Другий параметр відображає рівень заряду від 1-100 %.

Третій параметр показує напругу живлення модуля у мілівольтах (mV).

```
// Retrieves the battery status from the GSM module
int getBatteryStatus() {
    gsmModule.println("AT+CBC"); // Send the command to request battery status
    unsigned long currentTime = millis(); // Get the current time
    String batteryResponse = ""; // Variable to store the battery response
    while (millis() - currentTime < waitTime) {
        if (gsmModule.available()) {
            batteryResponse += gsmModule.readStringUntil('\n'); // Read the response from the GSM module
        }
    }
    if (batteryResponse.indexOf("+CBC:") >= 0) {
        // Extract the battery level from the response
        const int commaIndex1 = batteryResponse.indexOf(','); // Find the first comma
        const int commaIndex2 = batteryResponse.indexOf(',', commaIndex1 + 1); // Find the second comma
        const String batteryLevelStr = batteryResponse.substring(commaIndex1 + 1, commaIndex2); // Extract the battery level string
        return batteryLevelStr.toInt(); // Convert the battery level string to an integer and return it
    } else {
        sendSMS("Unable to read battery level"); // Send an SMS notification if unable to read battery level
        return -1; // Return -1 to indicate an error
    }
}
```

Рисунок 3.9 – Функція перевірка стану батареї

### 3.2.4 Реалізація функції обробки СМС-команд

Функція обробки СМС-команд (див. рис. 3.10) виконує обробку отриманих SMS-повідомлень на мікроконтролері. Вона перевіряє наявність нових SMS-повідомлень, отримує номер телефону відправника та текст повідомлення, і в залежності від отриманої команди виконує певні дії.

```

//Checks for incoming SMS messages and processes the commands.
void processSMSCommands() {
    // Check if there is data available to read from the GSM module
    if (gsmModule.available()) {
        String receivedData = gsmModule.readStringUntil('\n'); // Read the data from the GSM module

        if (receivedData.startsWith("+CMT:")) {
            // Extract the sender number from the receivedData string
            int firstIndex = receivedData.indexOf(' ');
            int secondIndex = receivedData.indexOf(' ', firstIndex + 1);
            String senderNumber = receivedData.substring(firstIndex + 1, secondIndex);
            senderNumber.trim(); // Remove leading/trailing whitespaces
            Serial.println("Admin number found: " + senderNumber);

            // Read the SMS text
            String sms = gsmModule.readStringUntil('\n');
            Serial.println("SMS text: " + sms);
            sms.trim();

            // Check if the sender number is in the list of adminNumbers
            bool isAdmin = false;
            for (int i = 0; i < sizeof(adminNumbers) / sizeof(adminNumbers[0]); i++) {
                if (senderNumber.equals(adminNumbers[i])) {
                    isAdmin = true;
                    break;
                }
            }

            if (isAdmin) {
                if (!sms.isEmpty()) {
                    // Process the SMS command based on its content
                    if (sms.indexOf("On") != -1) {
                        armAlarm(); // Turn on the alarm
                    } else if (sms.indexOf("Off") != -1) {
                        disarmAlarm(); // Turn off the alarm
                    } else if (sms.indexOf("Battery") != -1) {
                        // Get the battery level and send it as an SMS response
                        int batteryLevel = getBatteryStatus();
                        if (!isnan(batteryLevel)) {
                            sendSMS("Battery level: " + String(batteryLevel) + " %");
                        } else {
                            sendSMS("Unable to retrieve battery level.");
                        }
                    } else {
                        sendSMS("Invalid input. Please enter 'On', 'Off', or 'Battery'."); // Send an SMS with an invalid input message
                    }
                }
            } else {
                sendSMS("Unauthorized sender. Phone: " + String(senderNumber)); // Sending SMS that the number is not authorized
            }
        }
    }
}
}

```

Рисунок 3.10 – Функція обробки СМС-команд

Основні кроки функції включають:

- а) перевірка наявності нових SMS-повідомлень;
- б) отримання номера телефону відправника та тексту повідомлення;
- в) перевірка, чи номер телефону відправника належить адміністратору;
- г) виконання відповідних дій залежно від отриманої команди:
  - 1) активація сигналізації;
  - 2) вимкнення сигналізації;
  - 3) отримання рівня заряду батареї;

д) відправка відповідного SMS-повідомлення.

Функція дозволяє виконувати різні дії відповідно до отриманих SMS-команд, забезпечуючи контроль над функціональністю мікроконтролера через SMS-повідомлення.

### 3.2.5 Реалізація функції відправки СМС модулем GSM

Функція sendSMS (див. рис. 3.11) реалізує відправку SMS-повідомлення з мікроконтролера до визначених адміністраторських номерів телефону.

```
// Function for sending SMS messages
void sendSMS(String message) {
    // Iterate through each administrator number to send the SMS message to all administrators.
    for (int i = 0; i < sizeof(adminNumbers) / sizeof(adminNumbers[0]); i++) {
        gsmModule.println("AT+CMGS=\"" + adminNumbers[i] + "\""); // Send command to set the SMS recipient
        delay(1000); // Delay 1 second for the response
        gsmModule.println(message); // Send the text of the SMS message
        delay(1000);
        gsmModule.println((char)26); // Send Ctrl+Z character to end the SMS message
        delay(100);
        while (gsmModule.available()) {
            Serial.write(gsmModule.read()); // Output the response from the GSM module to Serial Monitor
        }
    }
}
```

Рисунок 3.11 – Функція відправки СМС

Вона приймає текстове повідомлення як вхідний параметр і виконує наступні кроки для відправки SMS-повідомлення:

- 1) ітерується по кожному номеру адміністратора в списку adminNumbers;
- 2) відправляє команду AT+CMGS до GSM-модуля з вказаним номером адміністратора, щоб встановити отримувача SMS-повідомлення [15];
- 3) відправляє текстове повідомлення через GSM-модуль за допомогою команди gsmModule.println(message);
- 4) перевіряє наявність відповіді від GSM-модуля та виводить її на

## Serial Monitor.

Таким чином, функція sendSMS дозволяє відправляти SMS-повідомлення до адміністраторів, повідомляючи їх про різні події або стан системи.

Повний лістинг коду наданий в Додатку А.

### 3.3 Тестування системи сигналізації

#### 3.3.1 Тест-кейс

Тест-кейс – це артефакт/документ, що описує сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації функції, що тестуємо, або її частини [17].

Тест-кейс описує кроки, які потрібно виконати, вхідні дані, очікувані результати та критерії оцінки результату (див. рис. 3.12).

ID	Назва	Опис	Умови виконання	Очікуваний результат
001	Включення сигналізації при подачі живлення	Перевірка включення системи	Система не підключена до живлення	Світлодіодна індикація на панелі системи запалюється. Адміністратор отримує SMS-повідомлення про включення системи.
002	Постановка на сигналізацію	Перевірка правильної постановки на охорону	Система сигналізації знаходиться в увімкненому стані. Система має належне живлення та підключені сенсори.	Світлодіодна індикація на панелі системи запалюється. Адміністратор отримує SMS-повідомлення про постановку на охорону. Немає спонтанних спрацьовувань сенсорів або сигналів тривоги.
003	Зняття з сигналізації	Перевірка правильного зняття системи сигналізації з охорони.	Система сигналізації знаходиться в поставленому на охорону стані.	Світлодіодна індикація на панелі системи згасає, показуючи зняття з сигналізації. Адміністратор отримує SMS-повідомлення про зняття з охорони.
004	Датчик руху	Перевіряємо правильну реакцію системи сигналізації на виявлення руху.	Система сигналізації знаходиться в поставленому на охорону стані.	Світлодіодна індикація на панелі системи показує спрацювання сигналізації. Сигнали тривоги активуються, що включає звуковий сигнал. Адміністратор отримує SMS-повідомлення про виявлення руху.
005	Перевірка батареї	Перевіряємо правильну роботу системи перевірки батареї в системі сигналізації.	Система сигналізації ввімкнена	Адміністратор отримує SMS-повідомлення про стан батареї.
006	Тестування критичного рівня батареї	Перевіряємо правильну реакцію системи сигналізації на досягнення критичного рівня заряду батареї.	Система сигналізації ввімкнена. Батарея системи має критичний рівень заряду.	Адміністратор отримує SMS-повідомлення про критичний стан батареї.
007	Перевірка обробки СМС з не зареєстрованого номеру	Перевірка, як система сигналізації обробляє отриману СМС повідомлення з номера, який не є адміністраторським.	Система сигналізації працює та знаходиться в активному стані. Надіслати СМС повідомлення на номер системи сигналізації з не авторизованого номеру	Система сигналізації повинна коректно обробити СМС повідомлення з не адміністраторського номеру. Адміністратор отримує SMS-повідомлення про не авторизований номер.

Рисунок 3.12 – Тест кейс функціоналу системи сигналізації

### 3.3.2 Контрольний список

Контрольний список (Check list) – це список, який містить ряд необхідних перевірок під час тестування програмного продукту . Чек-лист може містити певні критерії, функції або сценарії, які потрібно перевірити або протестувати (див. рис. 3.13). Він може містити також додаткові вказівки або рекомендації щодо проведення тестування [16].

Перевірка	Результат
<b>Функціонал системи сигналізації</b>	
Включення сигналізації при подачі живлення	passed
Постановка на сигналізацію	passed
Зняття з сигналізації	passed
Перевірка датчика руху	passed
Перевірка статусу батареї	passed
Тестування критичного рівня батареї	passed
<b>Функціонал користувача</b>	
Перевірка обробки СМС з номера адміністратора	passed
Перевірка обробки СМС з не зареєстрованого номеру	passed

Рисунок 3.13 – Чек-лист функціоналу

Результати тестування системи сигналізації підтверджують її успішну працездатність і відповідність заданим вимогам. Усі функціональності, такі як включення та виключення сигналізації, реагування на рух та перевірка рівня батареї, були протестовані успішно. Система працює належним чином і забезпечує надійний захист об'єкту.



## ВИСНОВКИ

В рамках даної кваліфікаційної роботи, була проведена глибока аналітична та дослідницька робота з метою розробки і вдосконалення GSM сигналізації з використанням NodeMcu. Основною метою роботи було створення ефективної системи безпеки, яка забезпечує надійний моніторинг та захист об'єктів у режимі реального часу.

Під час розробки були використані сучасні технології, що дозволило реалізувати функціональність системи, включаючи сповіщення про спрацювання сигналізації, керування через СМС та моніторинг стану батареї.

Результати тестування підтвердили працездатність системи та відповідність її функціональних вимог. Система успішно реагує на включення та виключення сигналізації, виявлення руху та перевірку рівня батареї. Крім того, система ефективно обробляє SMS-повідомлення та надсилає сповіщення адміністраторам.

Також, варто зазначити, що розроблена GSM сигналізація має потенціал для подальшого вдосконалення. Деякі можливості для покращення включають розширення функціональності системи, наприклад, додавання можливості віддаленого керування через мобільний додаток. Також можна розглянути можливість додавання додаткових сенсорів, які можуть виявляти інші види небезпеки, наприклад, пожежу або протікання води.

Отже, розроблена GSM сигналізація з використанням NodeMcu є перспективним рішенням для захисту майна та забезпечення безпеки користувача, а подальше вдосконалення системи може покращити її функціональність і використання.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Як працює охоронна сигналізація. URL: <https://bezpeka.club/yak-pratsyuye-ohoronna-sygnalizatsiya/> (дата звернення: 09.03.2023).
2. Дерев'янко О. А., Антошкін О. А., Бондаренко С. М., Христич В. В. Системи пожежної та охоронної сигналізації: текст лекцій. Х.: УЦЗУ, 2008. С. 54-58.
3. What is GSM. URL: <https://www.electronics-notes.com/articles/connectivity/2g-gsm/basics-introduction.php> (дата звернення: 11.03.2023).
4. Нефедов В. І. Теорія електрозв'язку. Київ : Магнолія, 2018. 495 с.
5. ESP8266 Arduino Core's documentation. URL: <https://arduino-esp8266.readthedocs.io/en/latest/> (дата звернення: 14.03.2023).
6. NodeMCU ESP8266 datasheet. 2020. URL: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet> (дата звернення: 14.03.2023).
7. Söderby K., Hylén J. Getting Started with Arduino IDE. URL: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2> (дата звернення: 17.03.2023).
8. Arduino IDE Installation 2023. URL: <https://docs.arduino.cc/software/ide-v1/tutorials/Windows> (дата звернення: 17.03.2023).
9. SIM800L Hardware Design. URL: [https://www.makerhero.com/img/files/download/Datasheet\\_SIM800L.pdf](https://www.makerhero.com/img/files/download/Datasheet_SIM800L.pdf) (дата звернення: 22.03.2023).
10. HC-SR501 PIR MOTION DETECTOR. URL: <https://datasheetspdf.com/pdf-file/775434/ETC/HC-SR501/1> (дата звернення: 23.03.2023).
11. KY-012 Active Piezo-Buzzer module. URL: <https://datasheetspdf.com/pdf-file/1402025/Joy-IT/KY-012/1> (дата звернення: 23.03.2023).
12. Світлодіодне освітлення: переваги, недоліки та сфера використання. URL:

<https://europan.ua/news/svitlodiodne-osvitlennja-perevagi-nedoliki-ta-sfera-vikoristannja/> (дата звернення: 23.03.2023).

13. Літій-іонний акумулятор. URL: [https://uk.wikipedia.org/wiki/Літій-іонний\\_акумулятор](https://uk.wikipedia.org/wiki/Літій-іонний_акумулятор) (дата звернення: 23.03.2023).
14. Бібліотека SoftwareSerial. URL: <https://doc.arduino.ua/ru/prog/SoftwareSerial> (дата звернення: 26.03.2023).
15. AT Commands Set. URL: [https://www.arduino.cc/en/uploads/Main/Quectel\\_M10\\_AT\\_commands.pdf](https://www.arduino.cc/en/uploads/Main/Quectel_M10_AT_commands.pdf) (дата звернення: 26.03.2023).
16. Що таке чеклісти та як з ними працювати. 2023. URL: <https://training.qatestlab.com/blog/technical-articles/work-with-checklist/> (дата звернення: 27.03.2023).
17. Види Тест Кейсів. Стани Тест Кейсів. Багтрекінгові системи. Серйозність і пріоритет дефекту. Життєвий цикл дефекту. URL: <https://qlearning.com.ua/theory/lectures/material/test-cases-bugtracking-systems-severity-priority/> (дата звернення: 28.03.2023).

## ДОДАТОК А

### Програмний код системи сигналізації

```

#include <SoftwareSerial.h>

SoftwareSerial gsmModule(D7, D8); // RX=D7, TX=D8 pins for GSM module

const int motionPin = D3; // Motion sensor pin, connected to D3
const int buzzerPin = D1; // Buzzer pin, connected to D1
const int ledPin = D0; // LED pin, connected to D0

const unsigned long waitTime = 5000; // Set the maximum waiting time for response
const unsigned long checkInterval = 12 * 60 * 60 * 1000; // Battery level check
interval of 12 hours
unsigned long lastBatteryCheckTime = 0;

bool isArmed = false; // Alarm status
bool isMotionDetected = false; // Motion sensor status

String adminNumbers[] = {"+380123456789"}; // Record administrator phone numbers,
you can add more than one

// Function for sending SMS messages
void sendSMS(String message) {
    // Iterate through each administrator number to send the SMS message to all
administrators.
    for (int i = 0; i < sizeof(adminNumbers) / sizeof(adminNumbers[0]); i++) {
        gsmModule.println("AT+CMGS=\"" + adminNumbers[i] + "\""); // Send command to
set the SMS recipient
        delay(1000); // Delay 1 second for the response
        gsmModule.println(message); // Send the text of the SMS message
        delay(1000);
        gsmModule.println((char)26); // Send Ctrl+Z character to end the SMS message
        delay(100);
        while (gsmModule.available()) {
            Serial.write(gsmModule.read()); // Output the response from the GSM module to

```

```

Serial Monitor
    }
}

//Initializes the system and sets up the necessary configurations.
void setup() {
    Serial.begin(9600); // Initialize the Serial communication
    delay(1000); // Delay 1 second for stability

    pinMode(motionPin, INPUT); // Set motionPin as input
    pinMode(ledPin, OUTPUT); // Set ledPin as output
    pinMode(buzzerPin, OUTPUT); // Set buzzerPin as output

    gsmModule.begin(9600); // Initialize the GSM module communication
    delay(1000);

    gsmModule.println("AT"); // Send AT command to the GSM module
    delay(1000);
    while (gsmModule.available()) {
        Serial.write(gsmModule.read()); // Print the response from the GSM module to
Serial Monitor
    }

    gsmModule.println("AT+CMGF=1"); // Set SMS text mode
    delay(1000);
    while (gsmModule.available()) {
        Serial.write(gsmModule.read());
    }

    gsmModule.println("AT+CNMI=1,2,0,0,0"); // Set up SMS notification mode
    delay(1000);
    while (gsmModule.available()) {
        Serial.write(gsmModule.read());
    }

    // Delete all SMS messages
    gsmModule.println("AT+CMGD=1,4"); // Delete all messages in all storage locations

```

```

delay(1000);
while (gsmModule.available()) {
    Serial.write(gsmModule.read());
}

    sendSMS("Alarm system initialized. Enter 'On' to arm the alarm, 'Off' to disarm,
and 'Battery' to check battery status.");
    // Send an SMS to notify the administrators about the initialization and provide
usage instructions.
}

void loop() {

    processSMSCommands(); // check for user input and process commands
    checkMotionSensor(); // check motion sensor and send SMS if motion detected

    // check battery level every 12 hours and send SMS if critically lowOff
    unsigned long currentTime = millis();
    if (currentTime - lastBatteryCheckTime >= checkInterval) {
        checkBattery();
        lastBatteryCheckTime = currentTime;
    }

}

//Checks for incoming SMS messages and processes the commands.
void processSMSCommands() {
    // Check if there is data available to read from the GSM module
    if (gsmModule.available()) {
        String receivedData = gsmModule.readStringUntil('\n'); // Read the data from the
GSM module

        if (receivedData.startsWith("+CMT:")) {
            // Extract the sender number from the receivedData string
            int firstIndex = receivedData.indexOf(' ');
            int secondIndex = receivedData.indexOf(' ', firstIndex + 1);

```

```

String senderNumber = receivedData.substring(firstIndex + 1, secondIndex);
senderNumber.trim(); // Remove leading/trailing whitespaces
Serial.println("Admin number found: " + senderNumber);

// Read the SMS text
String sms = gsmModule.readStringUntil('\n');
Serial.println("SMS text: " + sms);
sms.trim();

// Check if the sender number is in the list of adminNumbers
bool isAdmin = false;
for (int i = 0; i < sizeof(adminNumbers) / sizeof(adminNumbers[0]); i++) {
  if (senderNumber.equals(adminNumbers[i])) {
    isAdmin = true;
    break;
  }
}

if (isAdmin) {
  if (!sms.isEmpty()) {
    // Process the SMS command based on its content
    if (sms.indexOf("On") != -1) {
      armAlarm(); // Turn on the alarm
    } else if (sms.indexOf("Off") != -1) {
      disarmAlarm(); // Turn off the alarm
    } else if (sms.indexOf("Battery") != -1) {
      // Get the battery level and send it as an SMS response
      int batteryLevel = getBatteryStatus();
      if (!isnan(batteryLevel)) {
        sendSMS("Battery level: " + String(batteryLevel) + " %");
      } else {
        sendSMS("Unable to retrieve battery level.");
      }
    } else {
      sendSMS("Invalid input. Please enter 'On', 'Off', or 'Battery'."); //
Send an SMS with an invalid input message
    }
  }
} else {
  sendSMS("Unauthorized sender. Phone: " + String(senderNumber)); // Sending

```

```

SMS that the number is not authorized
    }
}
}

//Arming function
void armAlarm() {
    isArmed = true; // Set the alarm status to armed
    digitalWrite(ledPin, HIGH); // Turn on the LED
    sendSMS("Alarm armed"); // Send an SMS notification that the alarm is armed
}

//Disarming function
void disarmAlarm() {
    isArmed = false; // Set the alarm status to disarmed
    digitalWrite(ledPin, LOW); // Turn off the LED
    digitalWrite(buzzerPin, LOW); // Turn off the buzzer
    isMotionDetected = false; // Reset the motion sensor status
    sendSMS("Alarm disarmed"); // Send an SMS notification that the alarm is
disarmed
}

// Checks the motion sensor and triggers the alarm if motion is detected
void checkMotionSensor() {
    // Check if the alarm is armed and the motion sensor is triggered
    if (isArmed && !isMotionDetected && digitalRead(motionPin) == HIGH) {
        // Motion sensor detected motion
        isMotionDetected = true; // Set the motion sensor status to true
        digitalWrite(buzzerPin, LOW); // Turn off the buzzer
        digitalWrite(ledPin, HIGH); // Turn on the LED
        sendSMS("Alarm! Motion detected"); // Send an SMS notification for motion
detection
    }
    // If motion is detected, blink the LED and the buzzer
    if (isMotionDetected) {
        digitalWrite(ledPin, !digitalRead(ledPin)); // Toggle the LED state

```



```

    delay(500);
    digitalWrite(buzzerPin, !digitalRead(buzzerPin)); // Toggle the buzzer state
    delay(100);
}
}

// Retrieves the battery status from the GSM module
int getBatteryStatus() {
    gsmModule.println("AT+CBC"); // Send the command to request battery status
    unsigned long currentTime = millis(); // Get the current time
    String batteryResponse = ""; // Variable to store the battery response
    while (millis() - currentTime < waitTime) {
        if (gsmModule.available()) {
            batteryResponse += gsmModule.readStringUntil('\n'); // Read the response
from the GSM module
        }
    }
    if (batteryResponse.indexOf("+CBC:") >= 0) {
        // Extract the battery level from the response
        const int commaIndex1 = batteryResponse.indexOf(','); // Find the first comma
        const int commaIndex2 = batteryResponse.indexOf(',', commaIndex1 + 1); // Find
the second comma
        const String batteryLevelStr = batteryResponse.substring(commaIndex1 + 1,
commaIndex2); // Extract the battery level string
        return batteryLevelStr.toInt(); // Convert the battery level string to an
integer and return it
    } else {
        sendSMS("Unable to read battery level"); // Send an SMS notification if unable
to read battery level
        return -1; // Return -1 to indicate an error
    }
}

// Checks the critical battery level at a specified interval.
void checkBattery() {
    int batteryLvl = getBatteryStatus(); // Get the current battery level
    if (batteryLvl >= 0 && batteryLvl < 10) {
        // Battery level is critically low

```

```
        sendSMS("Battery level is critically low at " + String(batteryLvl) + " %.  
Please recharge the device.");  
    }  
}
```