

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач відділення
комп'ютерних технологій
Наталія СТЕФУРАК / _____ /
(підпис)
« ____ » _____ 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи
освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 123 «Комп'ютерна інженерія»
на тему:
«Система керування дверним замком»

Студент групи КІ-41

Станіслав ЛАЗУТА

(підпис)

Керівник роботи

Василь ПАВЛЮС

(підпис)

Консультанти:
з техніко-економічного
обґрунтування

Любов МЕЛЕНЧУК

(підпис)

Нормоконтролер

Ольга СЛЄПЦОВА

(підпис)

Тернопіль – 2024

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням
комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

« ____ » _____ підпис
2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

студенту Лазуті Станіславу Андрійовичу

(прізвище, ім'я та по-батькові студента)

1. Тема роботи: «Система керування дверним замком», затверджена наказом по коледжу від «27» листопада 2023р., №234а-н.
2. Термін здачі студентом завершеної роботи: «28» червня 2024 р.
3. Вихідні дані до роботи: актуальні технології та засоби керування дверними замками, наявні рішення на ринку, технічне завдання.
4. Перелік питань, які повинні бути розроблені в роботі:
 - а) основна частина дослідження предметної області, формалізація вимог до системи, проектування структури та алгоритму роботи системи, реалізація та тестування системи.
 - б) техніко-економічне обґрунтування аналіз ринку наявних рішень, дослідження кошторису реалізації системи, підрахунок загальної вартості розробки системи.

5. Перелік графічного матеріалу: структурна схема системи, блок-схема алгоритму роботи системи, схема з'єднання компонентів, електрична принципова схема.

6. Консультанти роботи:

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	Меленчук Л.І. вчена ступінь, звання		
	П.І.П. консультанта		

КАЛЕНДАРНИЙ ПЛАН
Кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до кваліфікаційної роботи.	23.11.2023	01.12.23
2.	Дослідження предметної області, огляд типових рішень.	02.12.2023	05.02.2024
3.	Дослідження технологій реалізації.	29.01.2024	07.02.2024
4.	Розробка функціональних вимог до системи та робота над її структурою.	08.02.2024	01.03.2024
5.	Проектування системи та підготовка відповідного графічного матеріалу.	03.03.2024	05.04.2024
6.	Вибір, встановлення та налаштування середовища реалізації.	18.03.2024	08.04.2024
7.	Реалізація системи та написання відповідного розділу кваліфікаційної роботи.	09.04.2024	09.05.2024
8.	Доопрацювання апаратної складової.	10.05.2024	16.05.2024
9.	Опрацювання економічного розділу та написання відповідного розділу кваліфікаційної роботи.	11.03.2024	03.05.2024
10.	Тестування системи та усунення недоліків.	17.05.2024	31.05.2024
11.	Робота над оформленням пояснювальної записки.	01.06.2024	18.06.2024
12.	Попередній захист кваліфікаційної роботи, доопрацювання.	18.06.2024	18.06.2024
13.	Підготовка до захисту кваліфікаційної роботи.	19.06.2024	27.06.2024
14.	Захист кваліфікаційної роботи.	28.06.2024	28.06.2024

Дата видачі «27»листопада 2023р. Керівник _____ / Василь ПАВЛЮС

Завдання прийняв до виконання _____ / Станіслав ЛАЗУТА

Реферат

Кваліфікаційна робота. Система керування дверним замком. 49 сторінок, 16 рисунків, 2 додатки, 5 джерел.

Метою даної роботи є розробка та реалізація системи керування дверним замком. Система повинна надавати можливість керувати замком як з допомогою RFID-ключів, так і віддалено через Інтернет за допомогою мобільного телефона чи планшета. Окрім того, система повинна зберігати історію спроб відкриття дверей з зазначенням точної дати та часу.

Для реалізації проєкту було використано мікроконтролер ESP8266, середовище розробки Arduino IDE для програмування мікроконтролера ESP, бібліотеки для роботи з периферійними пристроями.

У роботі розглянуто процес розробки та реалізації системи керування дверним замком, представлено її ключові характеристики та переваги, а також надано рекомендації щодо її практичного застосування.

КЕРУВАННЯ ДВЕРНИМ ЗАМКОМ, ВІДДАЛЕНЕ КЕРУВАННЯ, RFID-МІТКА, СОЛЕНОЇД, ESP8266, ARDUINO, ІНТЕРНЕТ РЕЧЕЙ.

Abstract

Graduation Project. Door Lock Control System. 49 pages, 16 figures, 2 appendices, 5 sources.

The aim of this work is to develop and implement a door lock control system. The system should allow the lock to be controlled both via RFID keys and remotely over the Internet using a mobile phone or tablet. Additionally, the system should store a history of door unlocking attempts, including the exact date and time.

To implement the project, the ESP8266 microcontroller, the Arduino IDE development environment for programming the ESP microcontroller, and libraries for working with peripheral devices were used.

The work discusses the process of developing and implementing the door lock control system, presents its key features and advantages, and provides recommendations for its practical application.

DOOR LOCK CONTROL, REMOTE CONTROL, RFID TAG, SOLENOID, ESP8266, ARDUINO, INTERNET OF THINGS.

ЗМІСТ

Скорочення та умовні позначки	6
Вступ.....	7
1 Аналіз предметної області та постановка завдання.....	8
1.1 Опис предметної області	8
1.2 Аналіз наявних рішень	9
1.3 Аналіз вимог до системи та постановка завдання	13
2 Проєктування системи.....	15
2.1 Проєктування структури системи	15
2.2 Проєктування алгоритму роботи системи.....	17
3 Реалізація та тестування системи	19
3.1 Вибір компонентів системи	19
3.2 Реалізація принципової електричної схеми пристрою.....	26
3.3 Реалізація програмного коду.....	29
3.4 Тестування роботи системи	32
4 Техніко-економічне обґрунтування проєкту.....	36
4.1 Аналіз ринку	36
4.2 Розрахунок витрат на реалізацію.....	38
4.3 Обґрунтування доцільності розробки	39
Висновки	41
Перелік джерел посилання	42
Додатки.....	43

					<i>КР.КІ 24.533.06.000 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Лазута С.А.			Система керування дверним замком	Лім.	Арк.
Перевір.		Павлюс В.П.					5
Рецензент		Посвятовська О.				Акрушів	
Н. Контр.		Слепцова О.Я.				49	
Затверд.		Стефурак Н.А.				ГФК. ВКТ. КІ-41	

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AC – Alternating Current

GPIO – General-Purpose Input/Output

I2C – Inter-Integrated Circuit

IDE – Integrated Development Environment

IoT – Internet of Things

OLED – Organic Light-Emitting Diode

RFID – Radio frequency identification

SPI – Serial Peripheral Interface

TTL – Transistor-Transistor Logic

UART – Universal Asynchronous Receiver/Transmitter

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У наш час технології стають невід’ємною частиною повсякденного життя, особливо коли мова йде про забезпечення комфорту та безпеки наших домівок. Інтернет речей (IoT) відіграє важливу роль у цьому процесі, дозволяючи нам керувати різними аспектами домашнього середовища з будь-якої точки світу.

Одним з важливих елементів в цій області є дверні замки, які ми щодня використовуємо для захисту нашого житла.

Метою даної роботи є розробка розумного дверного замка, що керуватиметься дистанційно через Інтернет, надаючи додаткові функції безпеки та зручності для користувача та зможе бути частиною комплексної системи домашньої безпеки.

Робота охоплює різноманітні аспекти розробки та реалізації розумного дверного замка, починаючи від аналізу вимог користувачів та архітектури системи до вибору та інтеграції необхідних компонентів та програмування пристрою.

Робота поєднує знання з різних галузей, таких як матеріали робототехнічних систем, програмування мікроконтролерів та технології Інтернету речей, з метою створення надійного та ефективного рішення.

Завданням роботи є не лише розробити новий пристрій, але й дослідити, як такі технології можуть бути впроваджені для підвищення якості нашого життя через забезпечення безпеки та зручності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Опис предметної області

У сучасному світі, де технології невпинно вдосконалюються, розумні пристрої стають необхідною частиною нашого повсякденного життя. Розумний дверний замок – це не просто захисний механізм, але і інноваційний продукт, який комбінує в собі безпеку та зручність використання. В розумному дверному замка, як правило, реалізована система, яка дозволяє користувачам забезпечити безпеку свого житла та зручно керувати доступом до приміщення. Завдяки технологіям Інтернету речей, дверний замок пов'язаний з Інтернетом, що відкриває нові можливості для віддаленого керування та моніторингу.

Типові функції розумного дверного замка включають:

1. Віддалене керування. Розумний дверний замок забезпечує користувачам можливість віддалено відкривати та закривати двері за допомогою мобільного застосунка або веб-інтерфейсу. Це зручно для ситуацій, коли користувачі відсутні вдома, але хочуть дати доступ гостям або кур'єрами.

2. Автоматичне відкривання. Додатковою функцією може бути автоматичне відкривання дверей при наближенні авторизованих осіб, наприклад, з використанням технологій Bluetooth або RFID-карток. Це забезпечує зручний та швидкий доступ для власників без необхідності введення паролю чи використання звичайного ключа.

3. Моніторинг доступу. Система ведення журналу доступу дозволяє користувачам відслідковувати коли і хто входив або виходив з будинку. Це корисно для забезпечення безпеки та вирішення ситуацій, пов'язаних з контролем доступу.

4. Повідомлення та сигналізація. Розумний дверний замок може надсилати повідомлення власнику про будь-які підозрілі або невдалі спроби входу.

5. Інтеграція з іншими системами. Можливість інтеграції з іншими системами, такими як системи безпеки, відеоспостереження та інші розумні

					КР.КІ 24.533.06.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

пристрої вдома, розширить функціональність та створить єдину екосистему контролю над житловим простором.

1.2 Аналіз наявних рішень

Перш ніж приступити до розробки системи розумного дверного замка варто ретельно проаналізувати вже існуючі рішення на ринку. У цьому розділі ми розглянемо та порівняємо три найпопулярніші та інноваційні програми в даній сфері:

- SmartThings.
- August Home.
- Nuki Smart Lock.

SmartThings

Система SmartThings від Samsung – є однією з найпоширеніших платформ для розумного дому, яка включає в себе також керування дверним замком. Вона дозволяє користувачам віддалено керувати доступом, встановлювати графіки роботи та отримувати сповіщення про стан дверей. SmartThings дозволяє користувачам віддалено керувати розумним дверним замком через спеціальний мобільний застосунок. Незалежно від того, де вони знаходяться, користувачі можуть відкривати та закривати двері, надаючи доступ або блокуючи його. Функція графіків роботи дозволяє встановлювати розклади доступу для різних користувачів. Наприклад, власники можуть автоматично встановлювати графік для певних періодів, або тимчасово надавати доступ гостям. Користувачі отримують повідомлення про стан дверей та всі події, пов'язані з розумним дверним замком. Це може включати інформацію про те, коли був відкритий або закритий дверний замок, чи були спроби проникнення.

SmartThings інтегрується з широким спектром розумних пристроїв, що робить його центральною точкою для керування всім розумним домом. Вона може включати системи освітлення, термостати, камери відеоспостереження та інші пристрої. мобільний застосунок SmartThings має інтуїтивно зрозумілий та

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

зручний інтерфейс (рис. 1.1), що спрощує процес керування розумним домом, забезпечуючи користувачам високий рівень контролю та безпеки в їхньому житловому просторі.

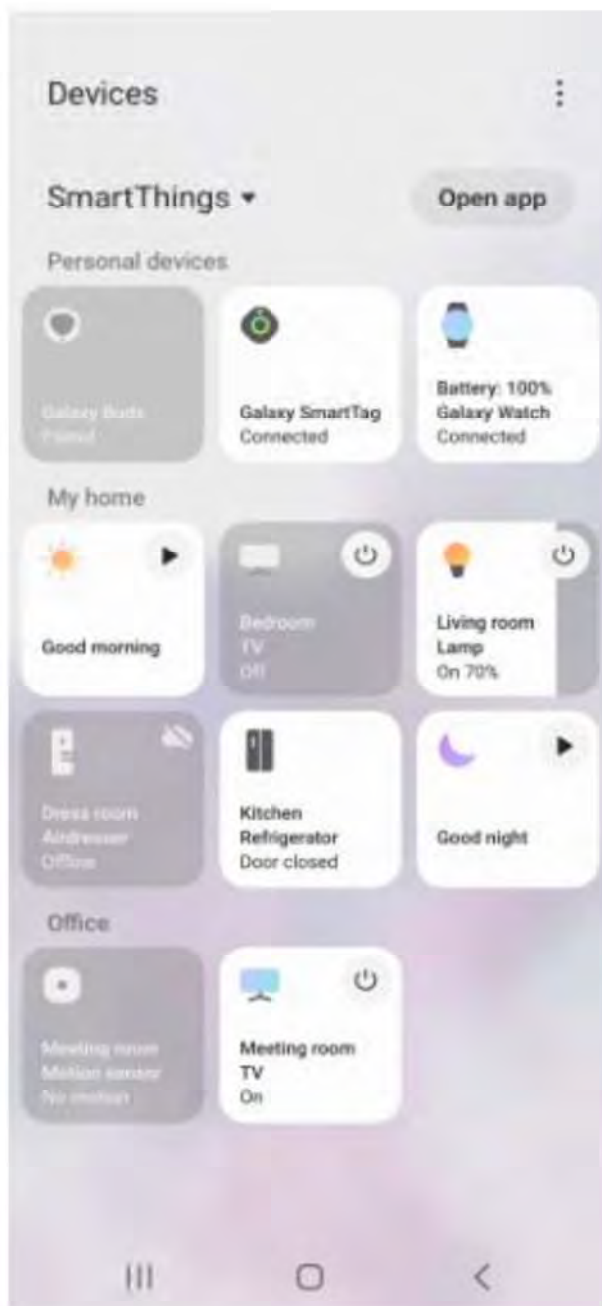


Рисунок 1.1 – Інтерфейс мобільного застосунку SmartThings

August Home

Система August Home представляє сучасний та надійний розумний дверний замок, який вирізняється високою функціональністю та інтеграцією з іншими популярними платформами розумного дому. Він розроблений з урахуванням широкої сумісності, підтримуючи популярні платформи, такі як

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Apple HomeKit і Amazon Alexa. Це дозволяє користувачам легко інтегрувати його в екосистему свого розумного будинку та керувати замком голосовими командами. Також August Home славиться своєю високою надійністю та рівнем безпеки – використання передових технологій шифрування та механізмів безпеки робить його відмінним вибором для захисту дому.

Зручний та інтуїтивно зрозумілий мобільний застосунок August Home (рис. 1.2) дозволяє віддалено керувати дверним замком з будь-якого місця. Користувачі можуть відкривати та закривати двері, слідкувати за їхнім станом і використовувати різні функції. Також є тимчасовий доступ, система повідомлень та журнал дій як і в SmartThings.

August Home є сучасним та надійним розумним дверним замком, який відповідає високим стандартам безпеки та функціональності. Його інтеграція з популярними платформами розумного дому та можливість контролю з використанням мобільного застосунка роблять його привабливим вибором для тих, хто цінує зручність та безпеку.

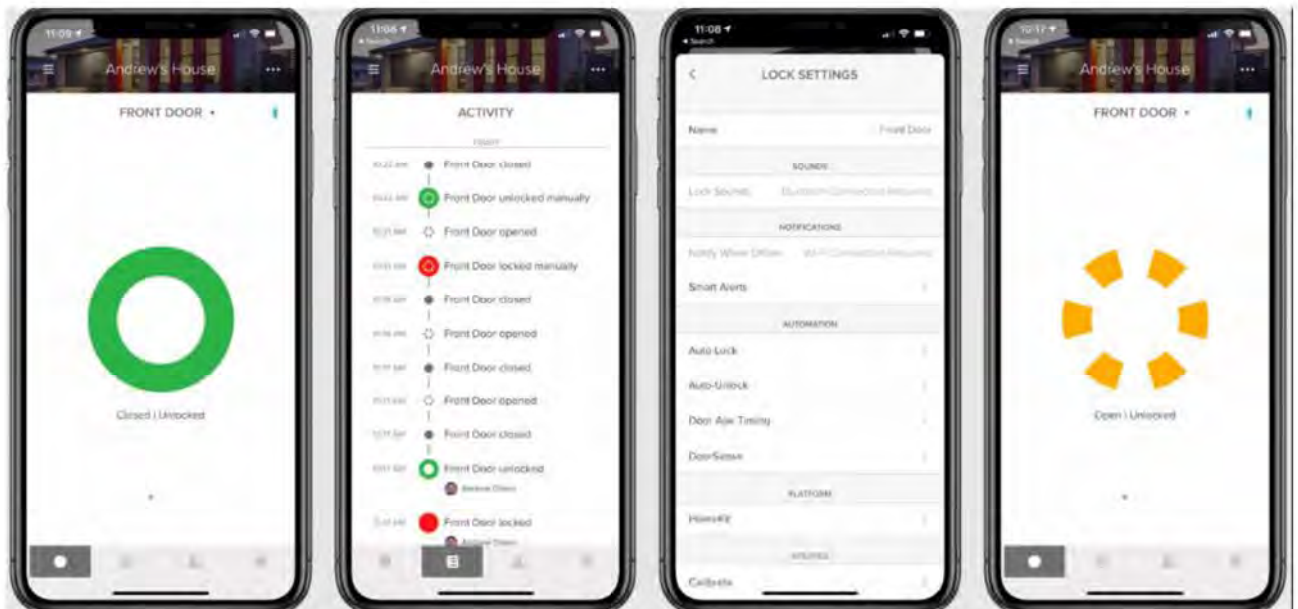


Рисунок 1.2 – Інтерфейс мобільного застосунку August Home

Nuki Smart Lock

Система Nuki Smart Lock являє собою компактний та передовий розумний дверний замок, який відрізняється своєю простотою в установці та високою ефективністю у керуванні доступом, Nuki Smart Lock визначається швидкістю та простотою встановлення. Компактний дизайн та інтуїтивно зрозумілі інструкції роблять процес установки максимально простим, що дозволяє користувачам швидко насолоджуватися всіма перевагами розумного дверного замка.

Також мобільний застосунок Nuki Smart дозволяє власникам віддалено відкривати та закривати двері за допомогою Bluetooth або WiFi. Це надає велику гнучкість та контроль над доступом до будинку.

Nuki Smart Lock розширює свої можливості через інтеграцію з іншими розумними системами. Це може включати взаємодію з розумними системами освітлення, опалення та безпеки, що робить його важливим елементом у розумному домі.

Мобільний застосунок Nuki Smart відрізняється інтуїтивно зрозумілим інтерфейсом, який спрощує взаємодію з дверним замком (рис. 1.3). Користувачі можуть легко налаштовувати параметри та контролювати стан замка.



Рисунок 1.3 – Інтерфейс мобільного застосунку Nuki Smart Lock

Порівняльний аналіз схожих продуктів

Тепер, коли кожна з програм була представлена попередньо окремо, варто їх порівняти за наступними критеріями: функціональність, інтеграція, безпека, зручність інтерфейсу, надійність та продуктивність, вартість, підтримка та оновлення (табл. 1.1).

Таблиця 1.1 – Порівняльна характеристика розумних замків

Критерії	SmartThings	August Home	Nuki Smart Lock
Функціональність	багато-функціональний	багато-функціональний	багато-функціональний
Інтеграція	можлива	можлива	можлива
Безпека	висока	висока	дуже висока
Зручність інтерфейсу	зручний та зрозумілий	зручний та зрозумілий	зручний та зрозумілий
Надійність та ефективність	надійний, ефективний	надійний, високоефективний	надійний, високоефективний,
Вартість	доступна	висока	доступна
Оновлення та підтримка	регулярні, активна	часті, хороша	часті, активна

1.3 Аналіз вимог до системи та постановка завдання

Провівши аналіз предметної області та здійснивши огляд наявних рішень пропонується реалізація власного проєкту – системи керування дверним замком.

Список практичних задач, які необхідно виконати в ході кваліфікаційної роботи, включає наступні:

– Розробити апаратну частину системи, включаючи безпосередньо електромеханічний механізм відкривання та закривання дверей, засоби зчитування безконтактних електронних ключів та засоби віддаленого доступу.

– Реалізувати програмне забезпечення, включаючи зручний та інтуїтивно зрозумілий користувацький інтерфейс керування системою через Інтернет.

- Забезпечити можливість збереження та аналізу хронологічних даних про відкривання та закривання дверей.
- Провести випробування та налаштування розробленої системи для оцінки її ефективності та надійності.
- Проаналізувати отримані результати та надати рекомендації щодо подальшого удосконалення системи.

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ СИСТЕМИ

Проектування системи розумного замка передбачає детальний аналіз потреб мешканців та вибір оптимальних технологій для забезпечення ефективного контролю та управління входом. У цьому розділі будуть розглянуті основні аспекти проектування системи, включаючи структуру системи та розробку інтерфейсу користувача.

2.1 Проектування структури системи

Для ефективного функціонування системи розумного замка потрібно організувати компоненти та встановити взаємодію між ними. Це включає визначення завдань кожного елемента, розробку механізму їх взаємодії, вибір методів передачі даних.

Після вибору необхідних компонентів системи розумного замка важливо скласти план їх взаємодії. Дані можуть передаватися від датчиків до контролера як проводами, так і через бездротовий зв'язок, після чого контролер обробляє отриману інформацію та виконує потрібні дії, такі як відкриття або закриття замка. Важливо також вибрати відповідні методи передачі даних, наприклад, через WiFi, для зв'язку між системою та користувачем.

Загалом, система передбачає наявність наступних компонентів:

1. Модуль керування (мікроконтролер), який буде центральним елементом системи та здійснюватиме управління всіма її компонентами, обробку даних, отриманих від RFID-зчитувача, та прийняття рішень щодо надання доступу.

2. Модуль безконтактного зчитування RFID-ключів, який зчитуватиме унікальні ідентифікаційні номери з RFID-карток або брелоків.

3. Модуль механічного блокування дверей (електрозамок), який керуватиметься мікроконтролером та відкриватиме або закриватиме двері в залежності від результату перевірки RFID-ключа.

4. Модуль відображення інформації, який відображатиме статус системи, інформацію про зчитані RFID-ключі, результати доступу, та інші повідомлення для користувача.

5. Модуль звукового сповіщення користувача, який видаватиме звукові сигнали для індикації різних подій, таких як успішне зчитування RFID-картки, відкриття замка тощо.

Структурна схема системи представлена на рисунку 2.1.

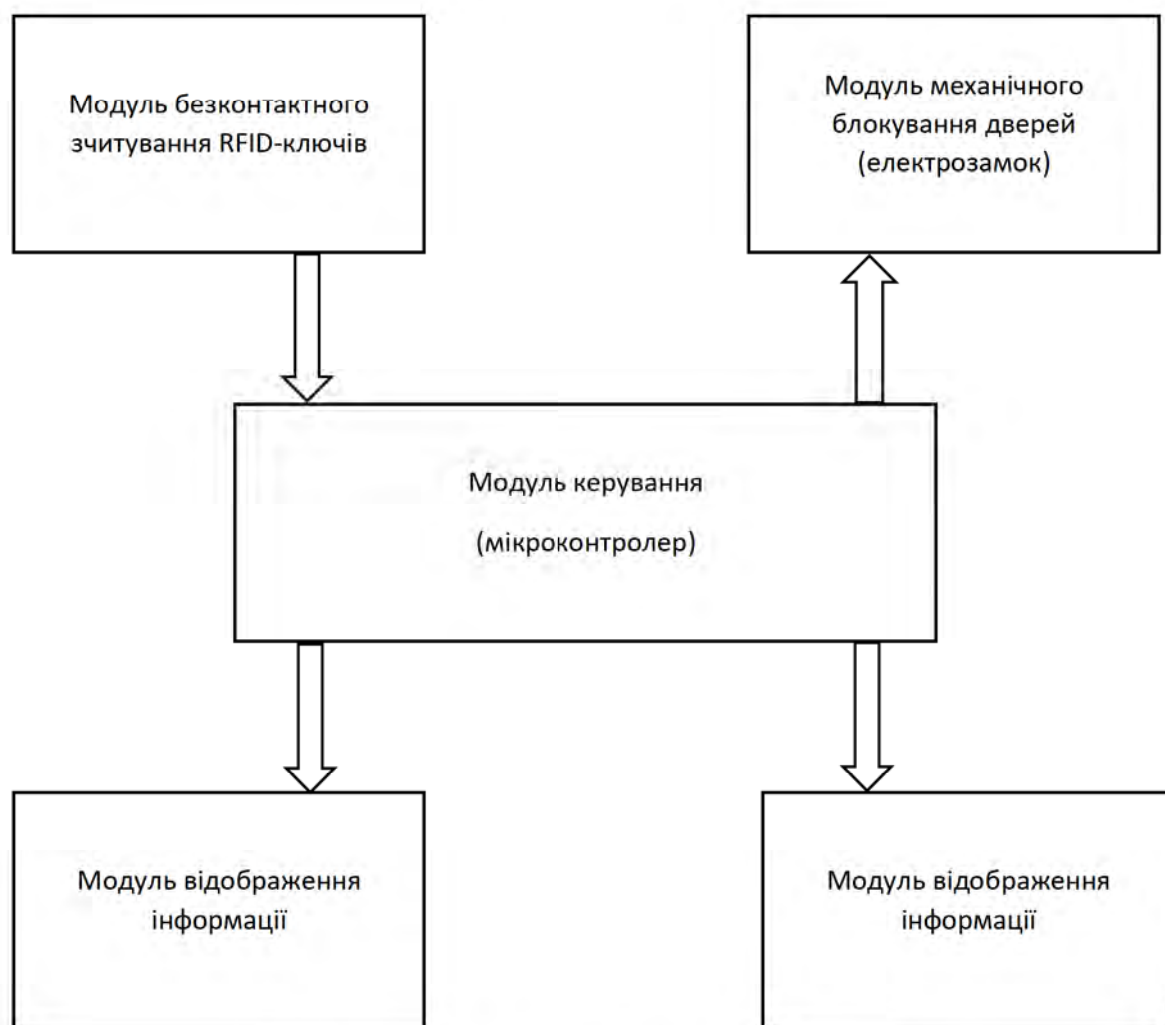


Рисунок 2.1 – Структурна схема системи

2.2 Проектування алгоритму роботи системи

Загалом, роботу системи можна розглядати як два паралельні процеси: робота з використанням RFID-міток та робота за допомогою браузера через веб-інтерфейс. Розглянемо ці процеси детальніше.

Робота з використанням RFID-міток:

1. Очікування нової картки.

- Система постійно сканує зону дії RFID-зчитувача на наявність нових карток.
- Зчитувач надсилає сигнал мікроконтролеру, коли виявляє нову картку.

2. Зчитування картки.

- Мікроконтролер отримує сигнал про виявлення нової картки.
- Зчитувач зчитує ID картки та передає його мікроконтролеру.

3. Перевірка прав доступу.

- Мікроконтролер порівнює отриманий ID з базою відомих карток.
- Якщо ID знайдено в базі:
 - Система вважає картку авторизованою.
 - Мікроконтролер активує реле, що відкриває замок.
- Якщо ID не знайдено в базі:
 - Система вважає картку неавторизованою.
 - Реле залишається неактивним і замок не відкривається.

4. Відображення результату на OLED-дисплеї.

- Система відображає наступну інформацію:
 - ID картки.
 - Ім'я користувача (якщо картка авторизована).
 - Результат перевірки (наявність права доступу).

5. Запис даних про спробу доступу.

- Система записує наступну інформацію про спробу доступу:
 - ID картки.

- Ім'я користувача (якщо відомо).
- Результат перевірки (наявність права доступу).
- Час спроби доступу.

Робота через веб-інтерфейс:

1. Відкриття інтерфейсу через браузер.
 - Користувач підключається до веб-інтерфейсу системи через браузер.
2. Відображення історії доступу.
 - Система виводить у веб-інтерфейс історію спроб доступу, включаючи:
 - ID картки.
 - Ім'я користувача.
 - Результат перевірки (наявність права доступу).
 - Час спроби доступу.
3. Відправка запиту на відкриття замка.
 - Користувач натискає кнопку для відкриття замка у веб-інтерфейсі.
4. Активація реле для відкриття замка.
 - Система через веб-інтерфейс отримує запит на відкриття замка.
 - Мікроконтролер активує реле, що відкриває замок на 5 секунд.
5. Запис даних про спробу доступу.
 - Система записує інформацію про спробу доступу через веб-інтерфейс:
 - Ім'я користувача (наприклад, «Smartphone»).
 - Результат перевірки (доступ дозволено).
 - Час спроби доступу.

Блок-схема алгоритму роботи системи представлена в додатку А.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Вибір компонентів системи

Вибір компонентів обґрунтований їхньою високою функціональністю, точністю та надійністю, що дозволить створити ефективну, надійну та бюджетно-ефективну систему управління доступом до будинку.

Мікроконтролер ESP8266, як і більшість мікроконтролерів, використовується для зчитування даних з датчиків та керування виконавчими механізмами.

Даний мікроконтролер часто використовується в платах розробки, які, окрім самого модуля ESP8266, мають вбудований USB-TTL конвертер, що дозволяє легко підключати мікроконтролер до комп'ютера через USB-порт для відлагодження та завантаження програмного коду в пам'ять мікроконтролера (рис. 3.1).

Переваги мікроконтролера ESP8266 [5]:

1. ESP8266 має значну продуктивність завдяки потужному процесору та достатній кількості вбудованих ресурсів. Це дозволяє ефективно виконувати завдання збору, обробки та передачі даних у реальному часі, що робить його ідеальним для різноманітних застосувань у сфері IoT.

2. Інтегрований WiFi: Вбудований модуль WiFi забезпечує можливість бездротового з'єднання з мережею WiFi, що робить ESP8266 ідеальним для віддаленого моніторингу та забезпечує зручність управління та обміну даними з іншими пристроями.

3. Розширені можливості програмування: ESP8266 може бути легко запрограмований з використанням мов C++, MicroPython чи Lua, що сприяє швидкій розробці та розширенню функціоналу пристрою.

Технічні характеристики ESP8266 [6]:

- Частота процесора: 80 МГц.
- Вбудована пам'ять: 4 Мб (Flash).
- Оперативна пам'ять (RAM): 80 Кб.

- Напруга живлення: 5 В.
- Вбудований WiFi.



Рисунок 3.1 – Мікроконтролер ESP8266 на платі розробки NodeMCU

RFID MFRC-522 – це модуль, який використовується для зчитування RFID-міток або карток. Він може взаємодіяти з мікроконтролерами, такими як Arduino або NodeMCU ESP8266, для реалізації систем контролю доступу або ідентифікації об'єктів (рис. 3.2).

Переваги MFRC-522:

1. Висока чутливість: Здатність зчитувати RFID-теги на відстані до кількох сантиметрів дозволяє ефективно використовувати модуль для різноманітних застосувань.
2. Широкий діапазон робочих частот: Модуль підтримує роботу на різних частотах, що дозволяє використовувати його у різних країнах та застосуваннях.
3. Простота використання: Інтерфейс SPI дозволяє легко інтегрувати модуль з мікроконтролерами та іншими пристроями.
4. Наявність бібліотек: Існують готові бібліотеки для роботи з модулем, що спрощує розробку програмного забезпечення для RFID-застосувань.

Технічні характеристики RFID MFRC-522 [3]:

- Робоча частота: 13.56 МГц.
- Зчитування на відстань до 5 см.
- Інтегровані антени.
- Підтримка стандартів: ISO/IEC 14443 та MIFARE.
- Напруга живлення: 3.3 В.

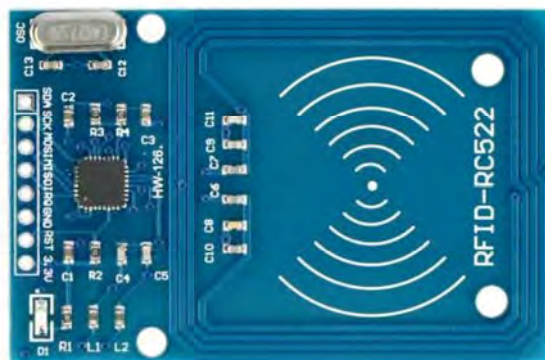


Рисунок 3.2 – Модуль RFID MFRC-522

Замок з електромагнітом – це пристрій на базі соленоїда, який використовується для керування замком дверей за допомогою електромагніту (рис. 3.3). Він дуже часто використовується для забезпечення безпеки та контролю доступу до приміщень або об’єктів.

Переваги електромагнітного замка:

1. Безпека: Забезпечує високий рівень безпеки, оскільки керується електронно.
2. Ефективність: Він ефективно виконує свою функцію, забезпечуючи надійний контроль доступу з мінімальними витратами на обслуговування.
3. Надійність: Електромагнітні замки мають високу надійність, що робить їх ідеальними для використання в системах безпеки.
4. Універсальність: Сумісний з багатьма системами контролю доступу та іншими електронними системами безпеки.

Технічні характеристики замка з електромагнітом:

- Напруга живлення: 12 – 220 В.
- Тип блокування: Електричний соленоїд.
- Монтаж: Вбудований у дверну конструкцію.



Рисунок 3.3 – Замок з електромагнітом

Реле – використовується для керування високовольтними електричними пристроями за допомогою низьковольтних сигналів з мікроконтролерів. Реле дозволяють ізолювати низьковольтну керуючу частину від високовольтної частини, забезпечуючи безпеку та надійність у керуванні електричними навантаженнями. Зазвичай, для зручності використання реле виконуються у вигляді готових модулів, які можуть мати незначні візуальні відмінності при ідентичних характеристиках (рис. 3.4).

Переваги реле:

1. Максимальна напруга та струм: Вихід реле може керувати високовольтними пристроями, зазвичай до 250V AC та 10A, залежно від типу реле.
2. Тип реле: Реле можуть бути нормально відкритими (NO – Normally Open) або нормально закритими (NC – Normally Closed).
3. Ізоляція: Використання в них оптопари забезпечує гальванічну ізоляцію між низьковольтною частиною і високовольтною частиною.

Технічні характеристики реле:

- Кількість каналів: 1 – 8.
- Напруга живлення: 3.3 – 5 В.



Рисунок 3.4 – Варіанти виконання модуля реле

OLED-дисплей SSD1306 – це популярний модуль для відображення інформації, який використовується в багатьох мікроконтролерних проектах. Цей дисплей характеризується низьким енергоспоживанням, високою контрастністю та можливістю відображення графічної та текстової інформації (рис. 3.5).

Переваги SSD1306 [2]:

1. Висока контрастність: OLED-дисплеї мають кращу контрастність у порівнянні з LCD-дисплеями, оскільки кожен піксель випромінює світло самостійно.
2. Низьке енергоспоживання: OLED-дисплеї споживають менше енергії, оскільки їм не потрібне підсвічування, як у випадку з LCD.
3. Широкий кут огляду: Кут огляду у OLED-дисплеїв наближається до 180 градусів, що забезпечує хорошу видимість з будь-якого кута.
4. Швидка реакція: OLED-дисплеї мають дуже швидкий час відгуку, що робить їх ідеальними для динамічного контенту.

Технічні характеристики SSD1306:

- Роздільна здатність: 128x64 або 128x32 пікселі.
- Розмір екрану: 0.96 або 1.3 дюйма.
- Підтримувані інтерфейси: I2C та SPI.
- Напруга живлення: 3.3 – 5 В.



Рисунок 3.4 – OLED-дисплей SSD1306

Бuzzer – це електромеханічний пристрій, який генерує звукові сигнали шляхом коливання мембранної пластини або іншого рухомого елемента. Вони використовуються для відтворення акустичних сигналів у різних пристроях, таких як годинники, будильники, електронні іграшки, сигналізаційні системи тощо. Бuzzer може бути активований за допомогою електричного сигналу, що змінюється у частоті або часі, щоб створювати різноманітні звукові ефекти.

Переваги бужера:

1. Простота використання: Бuzzer – це простий пристрій, який легко підключати та використовувати в різних пристроях.
2. Ефективність: Він здатен створювати чіткі звукові сигнали з низьким енергоспоживанням.
3. Надійність: Бuzzer має тривалий термін служби і відмінну стійкість до зовнішніх впливів.
4. Універсальність: Здатність генерувати різноманітні звукові сигнали робить їх корисними для різноманітних застосувань, від сигналізації до розважальних пристроїв.

Технічні характеристики бужера:

- Напруга живлення: 3.3 – 5 В.
- Потужність: 0.5 – 2 Вт.
- Робоча частота: 2 – 5 кГц.
- Опір: 16 – 32 Ом.



Рисунок 3.5 – Модуль бузера

Понижуючий стабілізатор живлення LM2596 – це електронний пристрій, що використовується для зниження напруги від джерела живлення до необхідного рівня (рис. 3.6). Він широко використовується в електронних схемах для живлення різних пристроїв і компонентів, таких як мікроконтролери, світлодіоди тощо [1].

Переваги LM2596:

1. Ефективність: LM2596 працює з високою ефективністю, що забезпечує ефективне перетворення електричної енергії без значних втрат.
2. Налаштування напруги: Дозволяє легко налаштувати вихідну напругу в межах встановлених параметрів.
3. Великий діапазон входних напруг: Підтримує широкий діапазон входних напруг, що робить його універсальним для різних джерел живлення.
4. Невеликі розміри: Компактний дизайн і малий розмір дозволяють легко використовувати його в різних електронних пристроях і схемах.

Технічні характеристики LM2596:

- Входна напруга: від 3.0 до 40 В.
- Вихідна напруга: від 1,25 до 37 В.
- Максимальний вихідний струм: до 3 А.
- Ефективність: до 92%.
- Тип вихідного струму: стабілізований.



Рисунок 3.6 – Понижуючий стабілізатор живлення LM2596

Отже, ESP8266 NodeMCU, RFID MFRC-522 та електромагнітний замок є оптимальною комбінацією для системи розумного замка. ESP8266 NodeMCU забезпечує підтримку WiFi, що дозволяє здійснювати віддалене керування замком. RFID MFRC-522 дозволяє здійснювати доступ до замка за допомогою RFID карток або брелоків. Електромагнітний замок забезпечує електронне блокування дверей з можливістю відкривання за допомогою ESP8266 NodeMCU.

На основі наведеного опису вимог до системи та обраного обладнання містить набір компонентів для вдалої демонстрації роботи системи (табл. 3.1).

Таблиця 3.1 – Перелік компонентів системи

№	Назва компонента	Кількість
1	NodeMCU ESP8266	1
2	Модуль RFID MFRC-522	1
3	Електромагнітний замок	1
4	Модуль реле	1
5	OLED-дисплей SSD1306	1
6	Бузер	1
7	Стабілізатор живлення LM2596	1

3.2 Реалізація принципової електричної схеми пристрою

Маючи повний перелік необхідних для реалізації системи компонентів, перейдемо до реалізації принципової електричної схеми пристрою. Скористаємось для цього програмою Fritzing.

Fritzing – це програмне забезпечення з відкритим вихідним кодом, яке призначене для проєктування електронних схем і друкованих РСВ-плат. Програма надає інтуїтивний інтерфейс, що дозволяє як початківцям, так і досвідченим інженерам швидко та ефективно створювати електронні проєкти. Користувачі можуть розміщувати компоненти на віртуальній макетній платі, з'єднувати їх дротами та додавати коментарі до схеми. Бібліотека компонентів,

яка періодично оновлюється, включає широкий асортимент елементів, таких як резистори, конденсатори, транзистори, мікроконтролери та багато інших.

Принципова електрична схема пристрою представлена на рисунку 3.7. Схема повністю зрозуміла та додаткових роз'яснень не потребує.

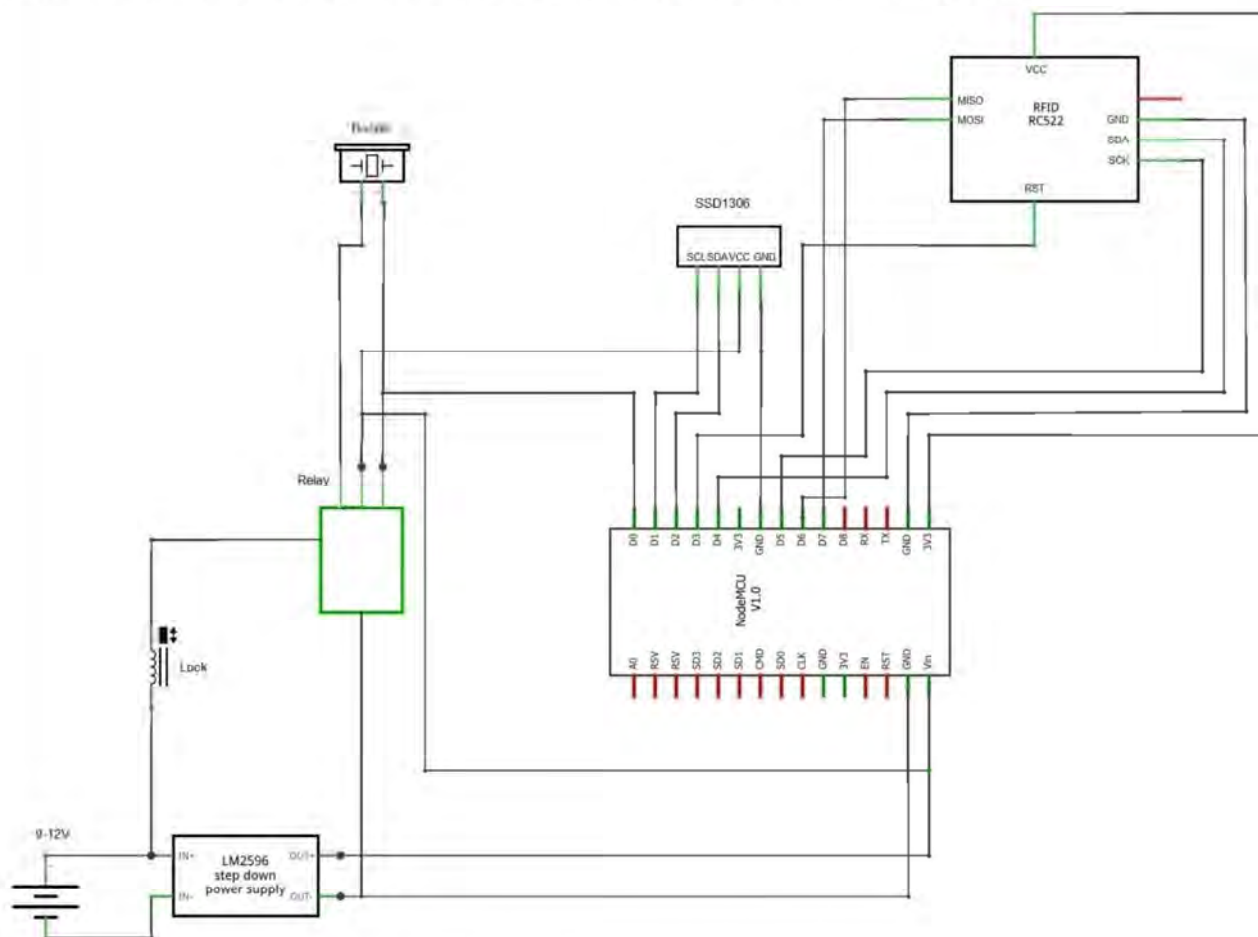


Рисунок 3.7 – Принципова електрична схема пристрою

На рисунку 3.8 представлена монтажна схема пристрою. В схемі використані всі згадані в описі компоненти. В якості джерела живлення для наочності використано батарею з напругою 9 вольт (типу «Крона»).

Змонтований на макетній платі пристрій представлений на рисунку 3.9.

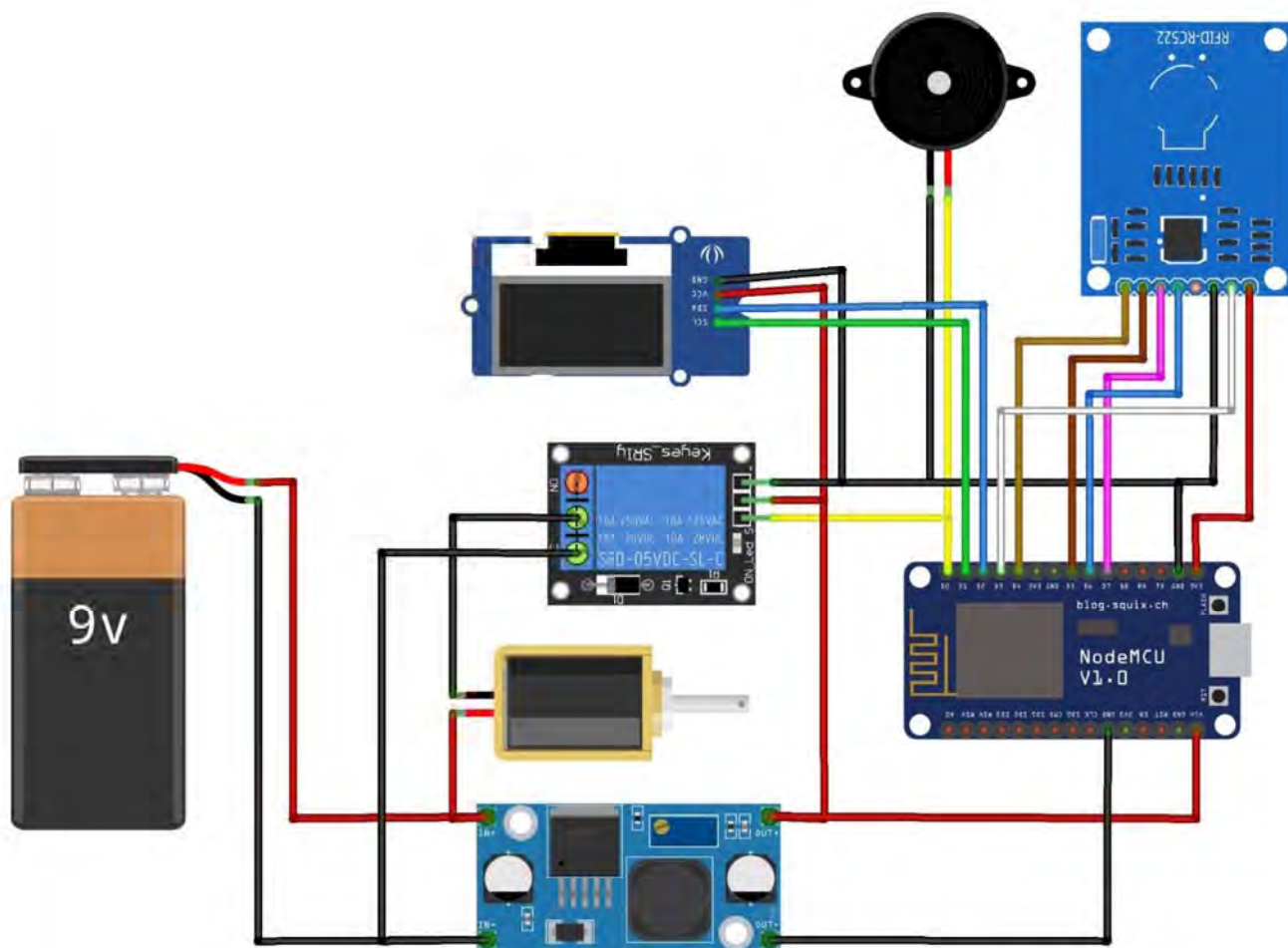


Рисунок 3.8 – Монтажна схема пристрою

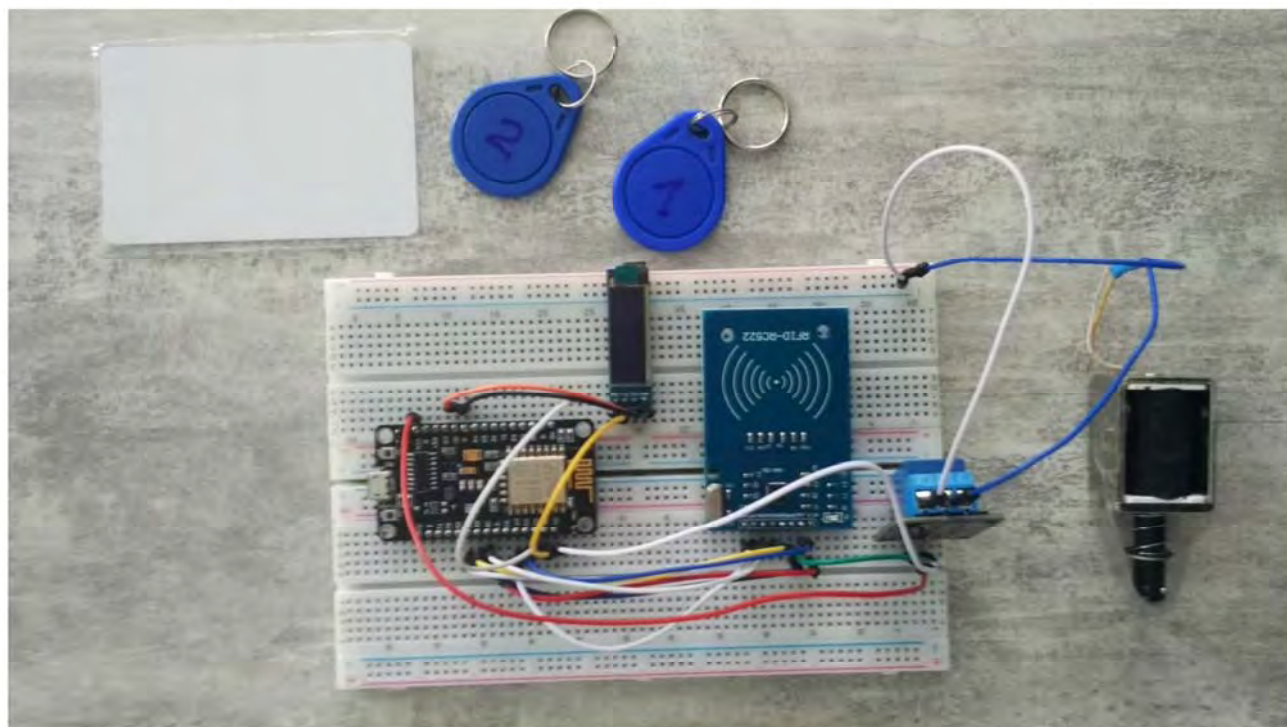


Рисунок 3.9 – Змонтований на макетній платі пристрій

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.533.06.000 ПЗ

Арк.

28

3.3 Реалізація програмного коду

Програмний код системи буде реалізований відповідно до описаного в пункті 2.2 алгоритму роботи системи.

Для реалізації програмного коду було обрано мову програмування C++ та середовище розробки Arduino IDE.

Вибір мови C++ зумовлений тим, що вона дозволяє ефективно працювати з апаратними ресурсами, що є критичним для мікроконтролерів. Мова забезпечує прямий доступ до пам'яті та апаратних регістрів, що дає змогу точно контролювати роботу мікроконтролера. Особливо важливим аргументом на її користь є й те, що для C++ існує багато бібліотек, зокрема для роботи з мікроконтролерами ESP8266. Це значно спрощує розробку, оскільки розробники можуть використовувати вже готові рішення для поширених завдань, таких як робота з WiFi, датчиками, дисплеями тощо.

Серед переваг середовища Arduino IDE варто відмітити наступні:

- має інтуїтивно зрозумілий інтерфейс, що робить процес програмування мікроконтролерів легким і швидким;
- дозволяє легко завантажувати код на мікроконтролер через USB, а також має вбудовані інструменти для відлагодження, що значно полегшує процес тестування і налагодження програм;
- підтримує мікроконтролери ESP8266, включаючи NodeMCU: встановлення додаткових пакетів та бібліотек для ESP8266 є простим і добре документованим;
- працює на Windows, macOS та Linux, що робить його доступним для користувачів різних операційних систем.

Процес написання коду розпочнемо з підключення необхідних бібліотек:

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
```



```
#include <WiFiUdp.h>
#include <NTPClient.h>
```

Далі здійснимо ініціалізацію OLED-екрану:

```
#define OLED_RESET -1
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define I2C_ADDRESS 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

Визначимо використовувані пін мікроконтролера для RFID-модуля та модуля реле:

```
#define RST_PIN 0 // RST для RC522 (D3)
#define SS_PIN 2 // SDA для RC522 (D4)
#define RELAY_PIN 16 // Пін для реле (D0)
```

Створимо екземпляр для RFID-модуля:

```
MFR522 mfrc522(SS_PIN, RST_PIN);
```

Створимо екземпляр вебсервера з портом 80:

```
ESP8266WebServer server(80);
```

Задамо константи для параметрів підключення до WiFi:

```
const char* ssid = "my_wifi";
const char* password = "my_password";
```

Задамо необхідні параметри для підключення до часового NTP-сервера та задамо часову зону UTC+3. Нам це знадобиться для фіксації точного часу відкриття дверного замка.

```
const long utcOffsetInSeconds = 10800;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "europe.pool.ntp.org",
utcOffsetInSeconds);
```

Створимо структуру даних для зберігання інформації про RFID-ключ, використаний для відкриття замка. Структура зберігатиме ID ключа, ім'я користувача, результату спроби доступу та поточного часу:

```
struct AccessAttempt {
    String cardID;
    String result;
    String user;
    unsigned long timestamp;
};
```

Створимо масив для зберігання записів про останні 10 спроб:

					КР.КІ 24.533.06.000 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

```
AccessAttempt accessHistory[10];
```

Для запису інформації про спробу доступу реалізуємо функцію `updateAccessHistory()`:

```
void updateAccessHistory(String cardID, String result, String
user, unsigned long timestamp) {
    accessHistory[historyIndex].cardID = cardID;
    accessHistory[historyIndex].result = result;
    accessHistory[historyIndex].user = user;
    accessHistory[historyIndex].timestamp = timestamp;
    historyIndex = (historyIndex + 1) % 10;
}
```

Для реалізації користувацького інтерфейсу реалізуємо функцію `handleRoot()`:

```
void handleRoot() {
    String html = "<html><head><meta charset=\"UTF-8\"><style>";
    html += "body { text-align: center; font-family: Arial, sans-
serif; }";
    html += "h1, h2 { color: #333; }";
    html += "table { margin: 0 auto; border-collapse: collapse;
width: 100%; max-width: 500px; }";
    html += "th, td { padding: 8px 12px; border: 1px solid #ddd; }";
    html += "button { padding: 10px 20px; font-size: 16px; cursor:
pointer; }";
    html += "</style></head><body><h1>Керування замком</h1>";
    html += "<p><button onclick=\"fetch('/toggleRelay').then(() =>
location.reload())\">Відкрити замок</button></p>";
    html += "<h2>Історія спроб</h2><table><tr><th>ID
карти</th><th>Користувач</th><th>Результат</th><th>Час</th></tr>";
    for (int i = 0; i < 5; i++) {
        int index = (historyIndex + i) % 5;
        if (accessHistory[index].cardID != "" ||
accessHistory[index].user == "Smartphone") {
            unsigned long eventTime = startTime +
accessHistory[index].timestamp / 1000;
            unsigned long hours = (eventTime / 3600) % 24;
            unsigned long minutes = (eventTime % 3600) / 60;
            String elapsedTime = String(hours) + ":" + (minutes < 10 ?
"0" : "") + String(minutes);
            html += "<tr><td>" + accessHistory[index].cardID +
"</td><td>" + accessHistory[index].user + "</td><td>" +
accessHistory[index].result + "</td><td>" + elapsedTime +
"</td></tr>";
        }
    }
    html += "</table></body></html>";
    server.send(200, "text/html", html);
}
```


Для керування дверним замком (за допомогою модуля реле) реалізуємо функцію `handleToggleRelay()`:

```
void handleToggleRelay() {  
    digitalWrite(RELAY_PIN, HIGH);  
    delay(5000);  
    digitalWrite(RELAY_PIN, LOW);  
    updateAccessHistory("", "Access allowed.", "Smartphone",  
millis());  
    server.send(200, "text/plain", "Relay toggled");  
}
```

Далі в функції `setup()` відбувається ініціалізація усіх підключених до мікроконтролера модулів, задання початкових значень змінних та підключення до NTP-сервера. Після цього в функції `loop()` циклічно відбувається зчитування коду RFID-ключа, його перевірка на валідність, відкриття дверного замка на 5 секунд в разі виявлення валідного ключа, та запис інформації про спробу в пам'ять мікроконтролера. Також в цій функції реалізується представлення збереженої інформації користувачу як за допомогою OLED-екрану, так і через вебінтерфейс за допомогою браузера.

Повна версія лістингу програмного коду представлена в додатку Б.

3.4 Тестування роботи системи

В процесі тестування роботи системи необхідно дослідити усі функції системи та зробити висновок про її функціональність. Для цього спершу необхідно скласти план тестування. Він включатиме наступні кроки:

1. Перевірити підключення пристрою до локальної WiFi-мережі.
2. Перевірити роботу OLED-екрану.
3. Перевірити зчитування пристроєм RFID-ключів.
4. Перевірити відкриття замка при використанні валідного ключа.
5. Перевірити ігнорування пристроєм невалідного ключа.
6. Перевірити підключення до системи за допомогою смартфона.
7. Перевірити роботу дверного замка за допомогою смартфона.

8. Перевірити відображення на смартфоні інформації про останні 10 спроб відкриття дверного замка.

Результати тестування відображені в таблиці 3.2 та на рисунках 3.10 – 3.12.

Таблиця 3.2 – Результати тестування роботи системи

№	Дія	Очікуваний результат	Фактичний результат
1	Підключення пристрою до мережі WiFi	Після увімкнення живлення пристрій підключився до WiFi	Після увімкнення живлення пристрій підключився до WiFi
2	Робота OLED-екрану	Після увімкнення на екрані з'явилося сповіщення System OK	Після увімкнення на екрані з'явилося сповіщення System OK
3	Зчитування RFID-ключів	При наведенні RFID-ключа на екрані відображено ID ключа та ім'я користувача	При наведенні RFID-ключа на екрані відображено ID ключа та ім'я користувача
4	Відкриття замка при використанні валідного ключа	При наведенні валідного RFID-ключа замок відкривається на 5 секунд	При наведенні валідного RFID-ключа замок відкривається на 5 секунд
5	Ігнорування пристроєм невалідного ключа	При наведенні невалідного RFID-ключа замок не відкривається	При наведенні невалідного RFID-ключа замок не відкривається
6	Підключення до системи за допомогою смартфона	При переході в браузері на відповідну адресу відображено інтерфейс користувача	При переході в браузері на відповідну адресу відображено інтерфейс користувача
7	Відкривання дверного замка за допомогою смартфона	При натисканні в браузері кнопки «Відкрити замок» замок відкривається на 5 секунд	При натисканні в браузері кнопки «Відкрити замок» замок відкривається на 5 секунд
8	Відображення на смартфоні інформації про останні 10 спроб відкриття замка	При переході в браузері на відповідну адресу відображено останні 10 спроб відкриття замка	При переході в браузері на відповідну адресу відображено останні 10 спроб відкриття замка

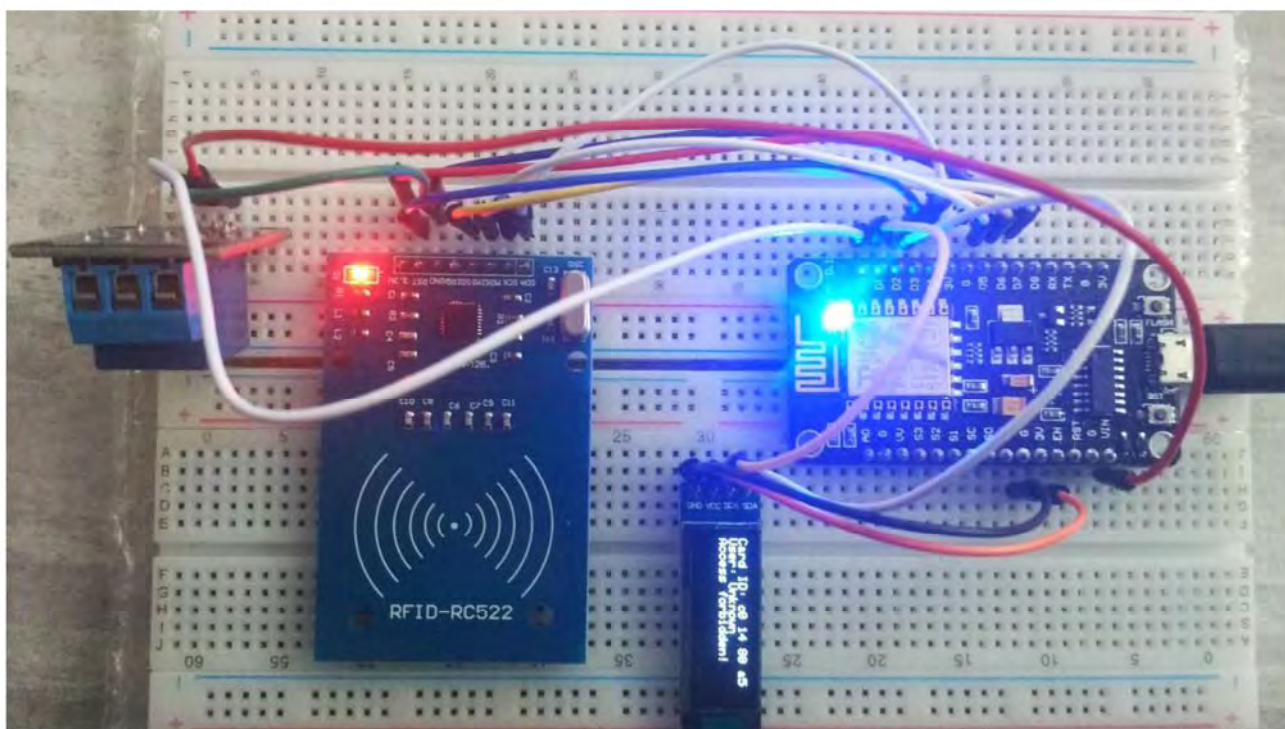


Рисунок 3.10 – Доступ заборонений при використанні невалідного ключа

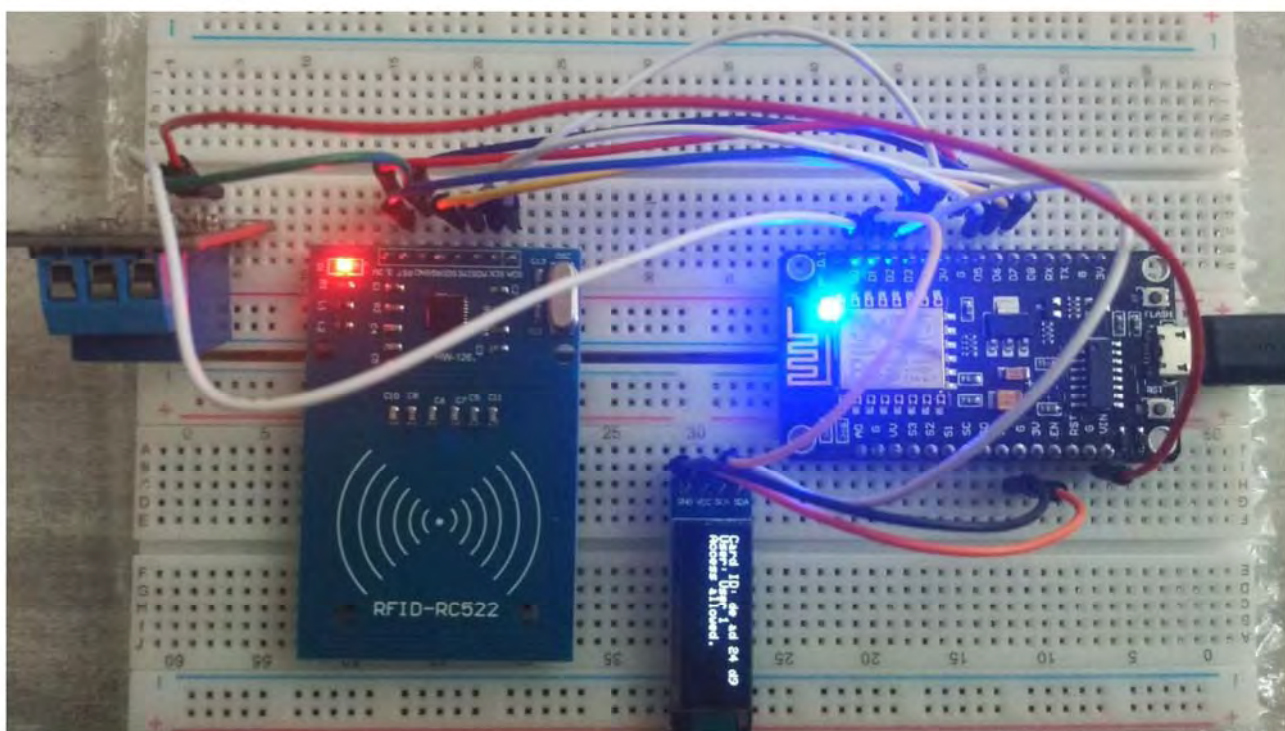


Рисунок 3.11 – Доступ наданий при використанні валідного ключа

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.533.06.000 ПЗ

Арк.

34

Керування замком

Відкрити замок

Історія спроб

ID карти	Користувач	Результат	Час
c0 14 80 a5	Unknown	Access forbidden!	15:58
	Smartphone	Access allowed.	16:00
b3 7a 6a 97	User 2	Access allowed.	16:01
de ad 24 d9	User 1	Access allowed.	16:02
c0 14 80 a5	Unknown	Access forbidden!	16:02

Рисунок 3.12 – Історія спроб відкриття замка

В процесі тестування роботи системи жодних недоліків не було виявлено. Система працює відповідно до поставленого завдання.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ПРОЄКТУ

4.1 Аналіз ринку

Системи керування дверними замками набирають популярності у сучасному світі завдяки зростанню попиту на розумні рішення для підвищення безпеки та зручності. Ці системи використовуються у житлових, комерційних і промислових об'єктах для забезпечення надійного доступу та моніторингу.

На ринку представлені різні типи розумних дверних замків, включаючи електронні замки з клавіатурою, біометричні замки (відбитки пальців), замки з Bluetooth та WiFi підключенням, а також RFID-замки.

Відомі виробники, такі як Yale, Schlage, і August, пропонують високоякісні та надійні продукти, які широко використовуються.

Зростаючий попит на розумні будинки та підвищення безпеки стимулює розвиток нових рішень у цій сфері.

Цінові діапазони на системи керування дверними замками залежать від їхніх функцій, технологій та брендів. Бюджетні рішення забезпечують базову безпеку та функціональність, середній сегмент пропонує більшу зручність та інтеграцію з розумними будинками, а преміум сегмент включає комплексні рішення з високим рівнем безпеки та багатофункціональністю, придатні для використання в корпоративних середовищах та розумних будинках з високими вимогами до безпеки.

Нижче наведено приблизні цінові діапазони для різних сегментів ринку.

1. Бюджетні рішення.

Простий електромагнітний замок з клавіатурою або картками

- Ціна: від 500 до 2000 грн.
- Характеристики: Прості електромагнітні замки, часто з клавіатурою для введення коду або з RFID-картами. Вони забезпечують базовий рівень безпеки та зручності. Підходять для встановлення в будинках, невеликих офісах або господарських приміщеннях.

2. Середній сегмент.

					КР.КІ 24.533.06.000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Розумні замки з Bluetooth або WiFi.

- Ціна: від 2000 до 5000 грн.
- Характеристики: Замки з можливістю підключення через Bluetooth або WiFi, що дозволяє керувати ними за допомогою смартфона. Мають додаткові функції, такі як тимчасові коди доступу, історія відкриттів, інтеграція з голосовими помічниками. Ідеально підходять для використання в розумних будинках.

Замки з біометричним доступом

- Ціна: від 5000 до 10000 грн.
- Характеристики: Замки з можливістю відкривання за відбитком пальця або через розпізнавання обличчя. Забезпечують високий рівень безпеки та зручності. Використовуються в житлових будинках, офісах та інших приміщеннях, де потрібен підвищений рівень захисту.

3. Преміум сегмент.

Комплексні системи безпеки з IoT.

- Ціна: від 10000 до 15000 грн і більше.
- Характеристики: Інтегровані системи, що включають різні типи доступу (ключі, карти, біометрія), з можливістю підключення до систем розумного будинку та управління через інтернет. Включають додаткові функції, такі як автоматизація сценаріїв, інтеграція з системами відеоспостереження та сигналізації. Використовуються в корпоративних офісах, готелях, великих житлових комплексах.

Приклади цін на ринку:

- Yale Assure Lock SL: близько 6000 грн.
- Schlage Encode Smart WiFi Deadbolt: близько 9000 грн.
- August Smart Lock Pro + Connect: близько 10000 грн.
- Samsung SHS-P718 Biometric Digital Door Lock: близько 15000 грн.

4.2 Розрахунок витрат на реалізацію

Для оцінки вартості реалізації системи керування дверним замком необхідно врахувати витрати на компоненти та заробітну плату інженера-програміста. Нижче наведено детальний розрахунок витрат.

Витрати на необхідні компоненти

Орієнтовна загальна вартість усіх необхідних компонентів для виготовлення однієї системи керування дверним замком становить приблизно 1150 грн. Детальний розрахунок вартості наведений у таблиці 4.1.

Таблиця 4.1 Вартість компонентів

Компонент	К-сть	Ціна за одиницю, грн	Загальна вартість, грн
NodeMCU ESP8266	1	200	200
Модуль RFID MFRC-522	1	120	120
Електромагнітний замок	1	600	600
Модуль реле	1	50	50
OLED-дисплей SSD1306	1	60	60
Бузер	1	20	20
Стабілізатор живлення LM2596	1	100	100
Разом			1150

Витрати на заробітну плату

Для реалізації проекту достатньо задіяти одного інженера-програміста, заробітна плата якого становить 30000 грн на місяць. Враховуючи податки та відрахування, розрахуємо загальні витрати на заробітну плату:

1. Брутто-зарплата: 30000 грн.
2. Податок на доходи фізичних осіб (ПДФО):
 - Ставка: 18%
 - Сума: $30000 * 0.18 = 5400$ грн.
3. Військовий збір:

- Ставка: 1.5%
- Сума: $30000 * 0.015 = 450$ грн.

4. Єдиний внесок:

- Ставка: 22%
- Сума: $30000 * 0.22 = 6600$ грн.

5. Загальні відрахування:

- ПДФО + Військовий збір: $5400 + 450 = 5850$ грн.

6. Сума до виплати:

- Брутто-зарплата – Загальні відрахування: $30000 - 5850 = 24150$ грн.

7. Загальні витрати на заробітну плату:

- Брутто-зарплата + Єдиний внесок: $30000 + 6600 = 36600$ грн.

Таким чином, витрати на заробітну плату складають 36600 грн.

4.2.3 Загальна вартість проекту

Загальна вартість реалізації проекту становить 37750грн., включаючи витрати на компоненти системи та заробітну плату інженера-програміста з урахуванням усіх податків і відрахувань (табл. 4.2).

Таблиця 4.2 – Загальна вартість проекту

Компоненти системи	1150 грн
Заробітна плата інженера	36600 грн
Разом	37750 грн

4.3 Обґрунтування доцільності розробки

Загальна вартість нашого проекту складає близько 37000 грн, тоді як вартість аналогічних рішень на ринку може сягати 15000 грн. На перший погляд може видатися, що реалізація власної системи недоцільна. Однак, варто врахувати, що зазначена вартість вказана для першого прототипу системи, часові затрати на який сягають близько одного місяця. Усі подальші системи виготовлятимуться за лічені дні, або навіть години, що суттєво скоротить їхню

вартість, яка не сильно перевищуватиме загальну вартість необхідних для її виготовлення компонентів.

Реалізація власної системи має й іншу важливу перевагу – вона дозволяє налаштувати систему відповідно до конкретних потреб користувача, що неможливо з багатьма готовими рішеннями на ринку. Це включає адаптацію дизайну інтерфейсу, інтеграцію з існуючими системами безпеки та автоматизації.

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно розроблено систему керування дверним замком на базі мікроконтролера NodeMCU ESP8266.

На початковому етапі було проведено детальний аналіз предметної області, який включав огляд сучасних систем керування дверними замками, їхніх функцій, характеристик та обмежень. Було визначено основні вимоги до системи та на їх основі сформульовано завдання проєкту.

Проєктування системи включало розробку структурної схеми системи та алгоритму її роботи.

На етапі реалізації проведено вибір компонентів, формування електричної схеми та її складання пристрою, написання та відлагодження програмного коду. Програмне забезпечення написано на мові програмування C++ у середовищі розробки Arduino IDE.

Тестування системи показало, що вона стабільно працює та відповідає поставленим завданням, включаючи функції керування замком за допомогою RFID-ключів, дистанційно через WiFi за допомогою браузера та відображення історії спроб відкриття замка.

Техніко-економічне обґрунтування включало аналіз ринку та розрахунок витрат на реалізацію системи. Даний аналіз показав, що розробка власної системи є виправданою.

Таким чином, розроблена система керування дверним замком є ефективним, надійним та економічно доцільним рішенням. Вона забезпечує зручність та гнучкість роботи, що робить її привабливим варіантом для приватного та подальшого комерційного використання.

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

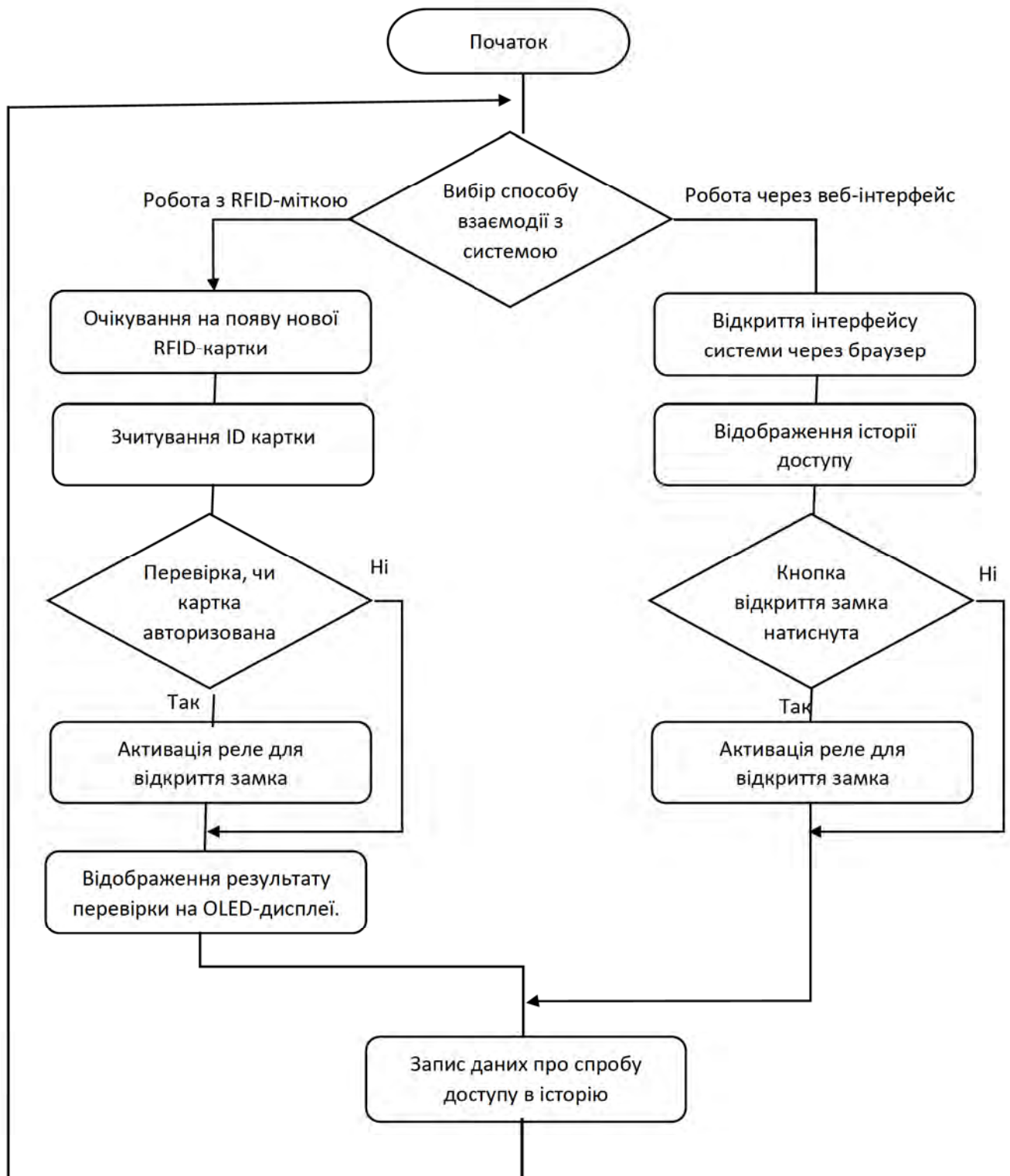
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. DC-DC понижуючий конвертер LM2596: веб-сайт. URL: <https://arduino.ua/prod650-dc-dc-ponijyuchii-konverter-lm2596-z-4-5-40v-do-3-35v> (дата звернення: 29.05.2024).
2. OLED-дисплей SSD1306: веб-сайт. URL: <https://arduino.ua/prod1795-oled-displei-modul-0-91-i2c-128x32-belii> (дата звернення: 18.03.2024).
3. RFID модуль RC522 з карткою доступу для Arduino: веб-сайт. URL: <https://arduino.ua/prod649-rfid-modul-rc522-s-kartochkoi-dostupa-dlya-arduino> (дата звернення: 19.04.2024).
4. Методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 123 «Комп'ютерна інженерія» освітньої програми «Інженерія Інтернету речей» / Павлюс В.П., Посвятовська О.Б., Кульчинська Н.З. – Галицький фаховий коледж імені В'ячеслава Чорновола, Тернопіль, 2023. 52с.
5. Мікроконтролер ESP8266: веб-сайт. URL: <https://uk.wikipedia.org/wiki/ESP8266> (дата звернення: 19.04.2024).
6. Налаштовуємо NodeMCU V3 Arduino IDE: веб-сайт. URL: <https://oxorona.com/nodemcu-v3-arduino> (дата звернення: 09.05.2024).

					КР.КІ 24.533.06.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

ДОДАТКИ

Додаток А. Блок-схема алгоритму роботи системи



Додаток Б. Лістинг програмного коду

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <WiFiUdp.h>
#include <NTPClient.h>

#define OLED_RESET -1
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define I2C_ADDRESS 0x3C // Адреса OLED-дисплея

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

#define RST_PIN 0 // RST для RC522 (D3)
#define SS_PIN 2 // SDA для RC522 (D4)
#define RELAY_PIN 16 // Пін для реле (D0)

MFRC522 mfrc522(SS_PIN, RST_PIN); // Створення екземпляра
MFRC522.

ESP8266WebServer server(80);

const char* ssid = "MY_WIFI";
const char* password = "MY_PASS";
const long utcOffsetInSeconds = 10800;
// Встановлення часового пояса UTC+3 (3600 секунд на годину)
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "europe.pool.ntp.org",
utcOffsetInSeconds);

unsigned long startTime = 0;

struct AccessAttempt {
    String cardID;
    String result;
```

```

String user;
unsigned long timestamp;
};

AccessAttempt accessHistory[10];
int historyIndex = 0;

void updateAccessHistory(String cardID, String result, String
user, unsigned long timestamp) {
    accessHistory[historyIndex].cardID = cardID;
    accessHistory[historyIndex].result = result;
    accessHistory[historyIndex].user = user;
    accessHistory[historyIndex].timestamp = timestamp;
    historyIndex = (historyIndex + 1) % 10;
}

void handleRoot() {
    String html = "<html><head><meta charset=\"UTF-8\"><style>";
    html += "body { text-align: center; font-family: Arial, sans-
serif; }";
    html += "h1, h2 { color: #333; }";
    html += "table { margin: 0 auto; border-collapse: collapse;
width: 100%; max-width: 500px; }";
    html += "th, td { padding: 8px 12px; border: 1px solid #ddd; }";
    html += "button { padding: 10px 20px; font-size: 16px; cursor:
pointer; }";
    html += "</style></head><body><h1>Керування замком</h1>";
    html += "<p><button onclick=\"fetch('/toggleRelay').then(() =>
location.reload())\">Відкрити замок</button></p>";
    html += "<h2>Історія спроб</h2><table><tr><th>ID
карти</th><th>Користувач</th><th>Результат</th><th>Час</th></tr>";
    for (int i = 0; i < 5; i++) {
        int index = (historyIndex + i) % 5;
        if (accessHistory[index].cardID != "" ||
accessHistory[index].user == "Smartphone") {
            unsigned long eventTime = startTime +
accessHistory[index].timestamp / 1000;
            unsigned long hours = (eventTime / 3600) % 24;
            unsigned long minutes = (eventTime % 3600) / 60;
            String elapsedTime = String(hours) + ":" + (minutes < 10 ?
"0" : "") + String(minutes);
            html += "<tr><td>" + accessHistory[index].cardID +
"</td><td>" + accessHistory[index].user + "</td><td>" +
accessHistory[index].result + "</td><td>" + elapsedTime +
"</td></tr>";
        }
    }
    html += "</table></body></html>";
    client.println(html);
}

```



```

    }
}
html += "</table></body></html>";
server.send(200, "text/html", html);
}

void handleToggleRelay() {
    digitalWrite(RELAY_PIN, HIGH);
    delay(5000);
    digitalWrite(RELAY_PIN, LOW);
    updateAccessHistory("", "Access allowed.", "Smartphone",
millis());
    server.send(200, "text/plain", "Relay toggled");
}

void setup() {
    Serial.begin(9600); // Ініціалізація Serial Monitor

    // Ініціалізація SPI шини та MFRC522
    SPI.begin();
    mfrc522.PCD_Init();
    Serial.println("Скануйте картку для відображення UID, SAK, типу
та даних блоків...");

    // Ініціалізація OLED дисплея без виведення логотипу Adafruit
    Wire.begin(4, 5); // SDA = GPIO 4, SCL = GPIO 5
    if (!display.begin(SSD1306_SWITCHCAPVCC, I2C_ADDRESS, false)) {
        Serial.println(F("Помилка SSD1306"));
        for (;;)
            ;
    }

    display.clearDisplay();
    display.setTextSize(2); // Встановлення розміру шрифту 2
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.print("System OK");
    display.display();
    delay(2000);

    // Ініціалізація реле
    pinMode(RELAY_PIN, OUTPUT);
    digitalWrite(RELAY_PIN, LOW);

```

					КР.КІ 24.533.06.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

```

// Підключення до WiFi
WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi..");
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
}
Serial.println("Connected");

// Відображення IP-адреси мікроконтролера в Serial Monitor
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());

// Налаштування веб-сервера
server.on("/", handleRoot);
server.on("/toggleRelay", handleToggleRelay);
server.begin();
Serial.println("HTTP server started");

// Підключення до сервера часу
timeClient.begin();
timeClient.update();

// Отримання початкового часу
startTime = timeClient.getEpochTime();
}

void loop() {
    // Перевірка веб-запитів
    server.handleClient();

    // Пошук нових карток
    if (!mfrc522.PICC_IsNewCardPresent()) {
        return;
    }

    // Вибір однієї з карток
    if (!mfrc522.PICC_ReadCardSerial()) {
        return;
    }

    // Виведення ID картки на OLED-дисплей

```



```

display.clearDisplay();
display.setTextSize(1); // Встановлення розміру шрифта 1
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);

// Отримання ID картки
String cardID = "";
for (byte i = 0; i < mfrc522.uid.size; i++) {
    cardID += (mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    cardID += (String(mfrc522.uid.uidByte[i], HEX));
}

// Виведення ID картки в перший рядок
display.print("Card ID:");
display.println(cardID);

// Перевірка, чи є ID картки відомим
bool knownCard = false;
String result;
String user;
if (cardID == " de ad 24 d9") {
    knownCard = true;
    result = "Access forbidden!";
    user = "User 1";
} else if (cardID == " b3 7a 6a 97") {
    knownCard = true;
    result = "Access allowed.";
    user = "User 2";
} else {
    result = "Access forbidden!";
    user = "Unknown";
}

// Виведення на OLED-дисплей тексту відповідно до ID картки
if (knownCard) {
    display.print("User: ");
    display.println(user);
    display.setCursor(0, 16);
    display.print(result);

    // Вмикаємо реле для відомих карток
    digitalWrite(RELAY_PIN, HIGH);
    delay(5000);
}

```

```

    digitalWrite(RELAY_PIN, LOW);
} else {
    display.print("User: Unknown");
    display.setCursor(0, 16);
    display.print(result);
}

// Оновлення історії доступу
unsigned long currentTime = millis();
updateAccessHistory(cardID, result, user, currentTime);

// Виведення ID картки
Serial.print("Card ID: ");
Serial.println(cardID);

display.display();

// Зупинка PICC
mfrc522.PICC_HaltA();
}

```

					<i>КР.КІ 24.533.06.000 ПЗ</i>	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		