

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ СХІДНОУКРАЇНСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ ФАКУЛЬТЕТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ КАФЕДРА
ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

До захисту
допускається
В.о. завідувач
кафедри
_____Лифар В.О.
« ____ » _____ 2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

_____ **бакалавр** _____

(освітньо-кваліфікаційний рівень)

НА ТЕМУ:

Кодовий замок з підключенням до мережі

Керівник роботи:

Марченко Д.М.

(підпис)

(ініціали, прізвище)

Студент:

Лободюк О.К.

(підпис)

(ініціали, прізвище)

Група:

КІ-17

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студента гр. КІ-17 Лободюк О.К.

Науковий керівник

Доцент, к.т.н.

Захожай О.І.

Оцінка наукового керівника:

Рецензент:

ПІБ, місто роботи, посада

Оцінка рецензента:

Кінцева оцінка за результатами захисту:

Голова ЕК

Захожай О.І.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Програмування та математики
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки 6.050102 «Комп'ютерна інженерія»

(шифр і назва)

Спеціальність

(шифр і назва)

ЗАТВЕРДЖУЮ:

В.о. завідувач кафедри

В.О. Лифар

«_____» _____ 20__ р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Лободюк Олег Костянтинович

(прізвище, ім'я, по батькові)

1. Тема роботи: **Кодовий замок з підключенням до мережі**

керівник проекту (роботи) Лифар В.О. кандидат технічних наук, доцент

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

затверджені наказом вищого навчального закладу від " " _____ 2021 р. №_

2. Строк подання студентом роботи 07 червня 2021

3. Вихідні дані до роботи Способи побудови систем обмеження доступу до приміщень. Розробка апаратного та програмного забезпечення.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналітичний огляд, розробка апаратного забезпечення, розробка програмного забезпечення. Висновки

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

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання 22 березня 2021 року

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Складання плану роботи	22.03.21 - 24.03.21	
2	Аналіз літератури	24.03.21 – 29.03.21	
3	Вивчення і підбирання матеріалу	29.03.21 – 20.04.21	
4	Написання розділів	20.04.21 – 25.05.21	
5	Оформлення пояснювальної записки	25.05.21 – 28.05.21	
6	Оформлення графічного матеріалу	28.05.21 – 03.06.21	
7	Підготовка доповіді і слайдів для презентації	03.06.21 – 07.06.21	

Студент

(Підпис)

_____ (прізвище та ініціали)

Науковий керівник

(Підпис)

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка включає 60 сторінок, 52 рисунків, 2 таблиць, 17 джерел за переліком посилань.

Ключові слова: NODEMCU V2 LUA, СОЛЕНОЇДНИЙ ЗАМОК, ВИДАЛЕНЕ УПРАВЛІННЯ, ЗЧИТУВАЧ RFID- МІТОК, BLUNK, ARDUINO IDE.

Мета роботи: створення електронного замка з підключенням, налаштуванням і управлінням через мережу.

Результат проектування : був спроектований електронний замок, що має можливість підключення налаштування і управління через мережу, а також із за допомогою RFID зчитувача також подачею звукового сигналу про відкриття замка або при спробі відкриття.

Додаток Blink використовується для зображення та управління з урахуванням усіх характеристик платформи, на якій побудовано пристрій.

Ця ідея відтворена фізично і буде показана в дії.

ЗМІСТ

ВСТУП.....	9
1.АНАЛІТИЧНИЙ ОГЛЯД	10
1.1 Формулювання актуальності, мети і завдань проекту	10
1.2 Аналіз початкових даних і відомих рішень	12
2.Проектний розділ	20
2.1 Розробка електричної схеми і вибір необхідних компонентів	20
2.2 Розробка конструкції пристрою	30
2.3 Розробка програмної частини пристрою	33
3. Практична реалізація проекту	40
3.1. Виготовлення системи	40
3.2. Перевірка і відладка програмної частини пристрою.....	42
Матеріальні витрати для реалізації проекту.....	47
Висновок.....	48
Список використовуваної літератури	49
Додаток А	51

ВИЗНАЧЕННЯ, ПОЗНАЧЕННЯ І СКОРОЧЕННЯ

У цій пояснювальній записці застосовують такі терміни з відповідними визначеннями:

NodeMCU v2 Lua-це платформа розробки на основі широко відомого чіпсета ESP8266 з відкритим початковим кодом.

Соленоїдний замок - у цих замках ригель наводиться в рух металевим сердечником, який втягується в соленоїд.

Зчитувач RFID – міток- пристрій, що дозволяє працювати з RFID- мітками. RFID (Radio Frequency Identification - радіочастотна ідентифікація).

Blynk - це служба Інтернету речей (IoT), розроблена для того, щоб зробити дистанційне керування і причитування цих датчиків з ваших пристроїв максимально швидким і простим.

IoT - Internet of Things, Інтернет речей.

Arduino IDE - інтегроване середовище розробки для Windows, MacOS і Linux, розроблена на Сі і С ++, призначена для створення і завантаження програм на Arduino- сумісні плати, а також на плати інших виробників. **Esp8266**- ESP8266 - мікроконтролер китайського виробника Espressif Systems з інтерфейсом Wi-Fi.

ОЗУ-вбудована оперативна пам'ять пристрою.

АЦП - Аналого-цифровий перетворювач.

GPIO - Інтерфейс введення/виведення загального призначення.

ДБЖ - Джерело безперебійного живлення

КМОН - структура або КМОН транзистор (комплементарна структура метал-оксид-напівпровідник

uidDec - Це отриманий "Card UID" / унікальний номер NFC карти

UID- це ідентифікатор, який був присвоєний виробником карти за допомогою контрольованої бази даних.

ВСТУП

Електронний замок — особливий електричний пристрій, потрібний з тією метою, щоб попередити доступ в закриті приміщення сторонніх осіб, або обмежити вихід з приміщення.

Система контролю приймає рішення про дозвіл на доступ в приміщення на базі сигналів від різних пристроїв: зчитувачів магнітних карт, штрих-кодів, датчиків контактної пам'яті, біометричних датчиків, матричної клавіатури, зчитувачів магнітних карт, віддаленого управління та інших різних датчиків.

У більшості випадків електронний замок є частиною важкою системою контролю доступу в приміщення. Як пристрій, які перешкоджають у доступі до приміщення, вживаються електромеханічні й електромагнітні запірні пристрої.

Впровадження мікроконтролера дозволить полегшити головні маніпуляції з системою контролю доступу і самим електронним замком. В рамках цієї роботи мається на увазі зробити, створити та налагодити маленьку що діє модель пристрою контролю доступу на Мікроконтролері управлінні.

Дана модель дозволить фактично показати роботу електричного замка і супутніх йому елементів.

1. АНАЛІТИЧНИЙ ОГЛЯД

1.1 Формулювання актуальності, мети і завдань проекту

В цей час електронні системи контролю доступу стають все більш поширеними у житті багатьох людей. Електронні замки використовуються там, де заборонено бути незнайомцем, наприклад, на складах компаній, у допоміжних приміщеннях торгових закладів та великих гіпермаркетах. Проте електронні системи контролю доступу є досить цінним обладнанням для пересічної людини. Що ж робити, якщо є рішення, які дозволять широкому колу мешканців використовувати електронні замки для різних цілей. Виходячи з вищесказаного, розробка функціонального пристрою для контролю доступу в приміщення на контролері мікроконтролера є життєво важливим завданням. В цей час електронні системи контролю доступу в приміщення стають все більш поширеними у житті багатьох людей. Електронні замки використовуються там, де заборонено бути незнайомцем, наприклад, на складах компаній, у допоміжних приміщеннях торгових закладів та великих гіпермаркетах. Не зважаючи на це, електронні системи контролю доступу є досить цінним обладнанням для пересічної людини. Що ж робити, якщо є рішення, які дозволять широкому колу мешканців використовувати електронні замки для різних цілей.

Відштовхуючись від вищесказаного, розробка функціонального облаштування контролю доступу в приміщення на мікроконтролері управлінні є життєво важливим завданням.

Завданням цієї роботи є: виготовлення працюючий макету облаштування контролю доступу на мікроконтролером управлінні.

Для досягнення цієї мети поставлені і виконані подальші завдання:

1. Огляд наявних рішень
2. Розробка схеми пристрою
3. Вибір потрібних компонент
4. Завдання методу роботи
5. Створення програми роботи пристрою

6. Виготовлення макету пристрою

7. Налаштування програми та макета

1.2 Аналіз початкових даних і відомих рішень

Сьогодні існує декілька варіантів замків, які в нашій звичайній свідомості не потрібний ключ, щоб відкривати або закривати двері. Є декілька видів замків - з магнітним ключем, які були представлені у вигляді брелків або карток, схожих на банківську карту, або з кодовим складом, що встановлюється безпосередньо на дверях, про що піде мова нижче.

Передові кодові замки можна систематизувати за декількома ознаками: за типом установки, способу управління механізмом замикання дверей, можливості змінювати кодову комбінацію.

За типом кодові замки розділяються на підвісні і врізні замки. Підвісні замки вживається в головному для сараїв, гаражів, складських приміщень, і в інших схожих варіантах, але для житлових будинків і квартир краще застосовувати врізні кодові замки(на рисунку 1.1), які надійніше за підвісних тому, що робочі механізми цього замку розміщуються усередині дверного полотна, захищаючи його від можливості розбору і злому зовні. Підвісні замки можна застосовувати в якості додаткового захисту.



Рисунок 1.1 - Замок врізного типу

За принципом управління запірним механізмом замки можна поділити на механічні, приклад показаний на рисунку 1.2, і електронні. У поточний час механічні замки починають втрачати свою привабливість, проте не можна без

коливань розповідати, що їх вже починають дуже тіснити електронні моделі, адже час механіки в нашій державі ще тривало триватиме, і важко точно повідомити, коли електроніка почне першенствування сфері замків. Не вважаючи на це, механічні замки мають свої недоліки, приміром, вони жвавіше виходять з ладу, в результаті незмінного їх використання, також суворим непогодженням буде те, що до цих замків досить просто підібрати відповідну комбінацію для відкриття, по співвідношенню з іншими видами.

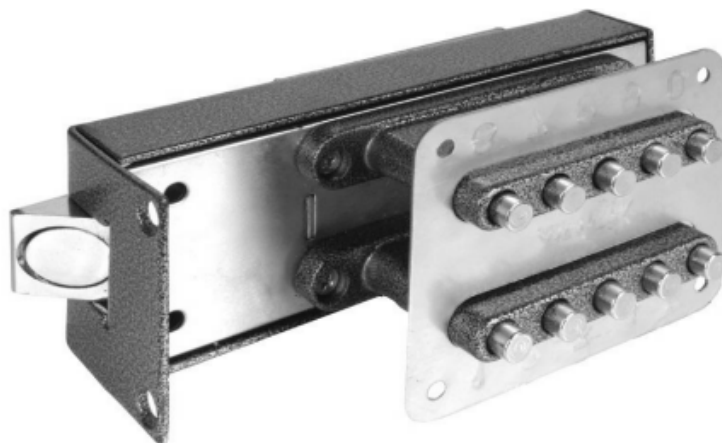


Рисунок 1.2 - Механічний кодовий замок

Механічні замки мають кнопковий кодовий механізм або поворотний. На практиці кнопкові замки демонструють себе найжахливіше. Незмінні натиснення на клавіші стирають їх верхній шар, і вони починають западати, що дає можливість хоч якій людині просто підібрати вірну комбінацію. Поворотний механізм, в зіставленні з кнопковим має безперечну перевагу - знайти, як і скільки разів і в яку сторону необхідно повернути ручку не вийти, не вважаючи на те, як затерті написи або як нерідко цим замком скористалися. Є також електронні кодові замки, що заслуговують на деяку увагу.

Електронні кодові замки мають одно значну перевагу перед механічними - блок управління може знаходитися у видаленні від головного пристрою. Його можна встановити у будь-якому місці - це база принципу замку - невидимки. Коли відсутня можливість бачити те, з чим доведеться мати справу, стає скрутно розкрити подібний пристрій. А також, сучасний кодовий замок - цей високотехнологічний пристрій, який управляється за допомогою процесора, а це мільйони різних композицій. Усі електронні кодові замки можна поділити на

подальші категорії:

1. Кнопочві кодові замки з електронним управлінням. Вони отримали найбільше поширення, але одночасно це уразлива група що замикає пристроїв. Причина, як і у разі механічних замків, та ж — клавіші, що стираються западаючи. У головному їх встановлюють там, де не вимагається зберігати особливо цінні предмети. Їх вживають для недопущення проникнення в приміщення, приміром в під'їзди, складські і підсобні приміщення в маленьких підприємствах або торгових точках. Майже завжди пульт управління і сам замок робляться одним блоком, але є моделі з роздільним оснащенням. З розвитком технологій кнопочві кодові замки отримали своє логічне продовження у вигляді замків з сенсорною панеллю. Достойність перед клавішами тут явні, але є суворя трудність з тим, що на мазких сенсорних екранах при натисненні залишаються відбитки пальців, які стають примітні під певним кутом огляду. Для уникнення злону композиції по відбитках, потрібно різні хитрощі, на які йдуть роби
2. Кодові замки з магнітним носієм композиції. Це надзвичайно надійні замки, розкрити які можна, якщо дістати сам магнітний носій. У цій ролі можуть виступати: пластмасова картка, маленький брелок, пульт видаленого управління, який передає код приймачу радіосигналом або сигналом в інфрачервоному діапазоні, просто перехоплюваний і декодований.
3. Комбіновані замки(Рисунок 1.3). Цей вид замків поширеніший на сьогодні. Потрапити усередину приміщення можна лише при використанні по черзі декількох різних що відкриває облаштувань, приміром, кодової композиції та пластмасової карти, що робить комбіновані замки надзвичайно надійною системою. Їх індивідуальністю буде те, що розблокувати двері нездатність лише за допомогою 1-го ключа, буде треба пройти усі почергові діяння, потрібні для відкриття, також те, що ключі від інших виробників ігноруватимуться при спробі їх використання.



Рисунок 1.3 - Електронний кодовий замок з сенсорною панеллю введення виробництва компанії Samsung

Облаштування електричного замку полягає 4 частин:

1. Запірний механізм. Щоб він відкрився або закрався, на нього подають маленький електричний імпульс. Коли кодова композиція співпаде з даною, замок розкриється.
2. Зчитувач коду або пульт управління. Це пристрій, що зчитує - електроніки управління в ньому немає. За допомогою цього пристрою вводиться код і направляється в блок управління.
3. Блок управління. Він розпоряджається питанням, подавати або не подавати імпульс для спрацювання запірної механізми. В основному замки закриваються автоматом як, в принципі, і хоч який інший механічний замок, що закривається
4. Джерело безперебійного живлення(ДБЖ). Наявність ДБЖ в таких замках обґрунтовано тим, що під час відключення електричної енергії потрапити у будинок буде проблемне, адже за відсутності електроенергії замки автоматом блокуються. З його допомогою кодовий електронний замок пропрацює автономною декілька діб. Зазвичай ДБЖ розміщується в потаємному місці, там же, де і блок управління.

У мережі Веб також можна знайти різні рішення схемотехніки в цілях

виконання електронних кодових замків, приміром, як на рисунку 1.4. Ця схема представляє з себе пристрій, який складається з : 2-ух КМОП мікросхем 561ЛА7 і однієї 561ЛЕ5, транзистори VT1 - VT3 - КТ361, або КТ3107, транзистор VT4 - КТ315, транзистор VT5 - КТ815. Вторинна обмотка трансформатора Т1 розрахована на 12 вольтів. Трансформатор Т1 вибирається достатньої потужності, яка забезпечує спрацьовування виконавчого пристрою, діоди VD3, - VD7 вибираються випрямні FR107, які забезпечують достатній струм виконавчого пристрою. Діоди VD8 - VD20 - малопотужні імпульсні КД521. В якості акумуляторної батареї вживається компактна лужна батарея, яка використовується в джерелах безперервного живлення. Набір коду робиться за допомогою кнопочової панелі SA1.

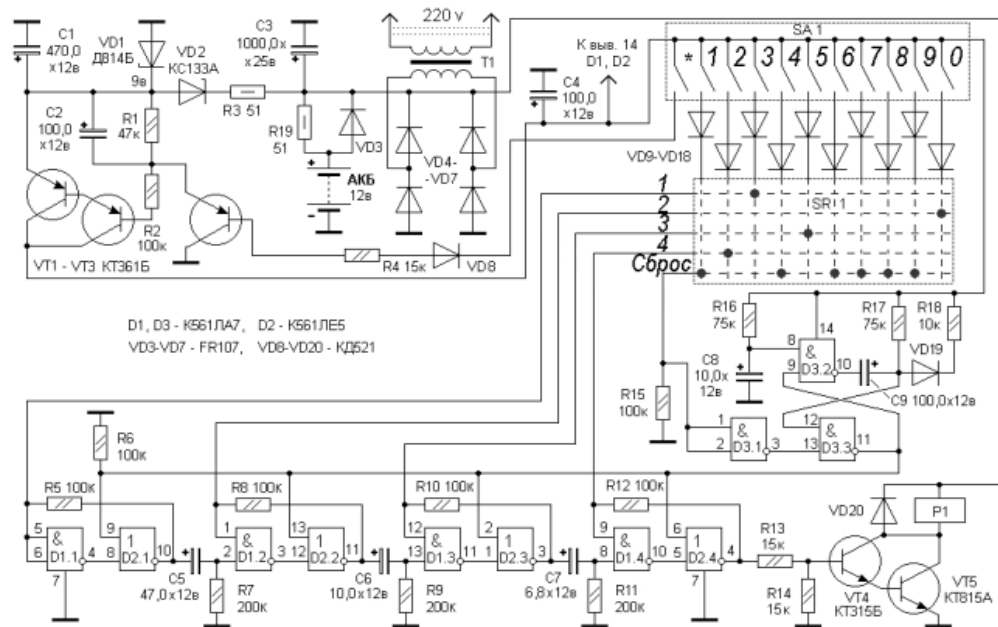


Рисунок 1.4 - Схема електрична принципова електронного кодового замку

Заявленими перевагами цієї схеми по співвідношенню з іншими рішеннями схемотехніки, доступними в мережі Веб являються : найвища захищеність від злому обманними клавішами, при натисненні на які замок блокується на деякий час, кнопочка панель розміщується окремо від головного пристрою, по цьому злом за допомогою вимірювальної апаратури стає неможливий, живлення від акумулятора на 12 вольтів, або від мережі змінного струму 220 вольтів.

Не вважаючи на це, яка була представлена вище схема кодового замку має свої недоліки як електронний замок : чутливість до електромагнітних завад,

важких метеорологічних умов, вимога до безперервного подання живлення. Проте основним чинником буде те, що для створення схожих рішень схемотехніки потрібно якнайглибші знання в схемотехніці та в мікро схемній логіці, потрібний чіткий підбір частин для правильної роботи, тому із-за усього переліченого вище досить важко вивчати цей тип замку в якості учбового знайомства. Потому в якості бази для цієї роботи був обраний електронний замок на мікроконтролерному управлінні.

Для того щоб утілити облаштування контролю доступу на мікроконтролером управлінні, в якості початкових даних були вибрані методи управління за допомогою радіочастотної ідентифікації, яка реалізується за допомогою спеціально розроблених для роботи з мікроконтролерами модулями радіочастотної ідентифікації. Розглянемо більше ретельно головні принципи організації радіочастотної ідентифікації.

RFID - радіочастотна ідентифікація є базовою і дешевою технологією, яка здійснює безпроводну передачу даних. Раніше ця розробка не так нерідко використовувалася в промисловості у зв'язку з відсутністю стандартизації в промислових компаніях. RFID - технології продуктивніші та надійніші, по співвідношенню з іншими. З RFID - технологією безпроводна автоматична ідентифікація сприймає надзвичайно специфічний вид: об'єкт, положення, або індивідом маркуватися неповторним ідентифікаційним кодом, що міститься у RFID, - мітці, яка якимсь чином прикріплена або втиснута в об'єкт. Радіочастотна ідентифікація являє собою не деякий продукт, а повноцінну систему. Звичайна RFID - система містить в собі три базисні елементи: RFID- мітку (трансбордер), зчитувач (трансфер, запитувач) і серверну програмку(інформаційну базу), яка просить підтримку комп'ютерної мережі. Програмне забезпечення вживається для управління, контролю, оперування, обробки, ведення обліку різних користувачів. Цифрова система блокування дверей робиться і управляється за допомогою RFID - зчитувача, який проводить перевірку й аутентифікацію користувача та автоматом відкривають двері. Крім того, він зберігає відомості про реєстрацію користувача. Надзвичайно принципово провести аутентифікацію

користувача до відкриття приміщення, що охороняється, і радіочастотна ідентифікація дозволяє зробити це. Система дозволяє користувачеві реєструватися у швидких, безпечних комфортних критеріях. Система блокування дверей відкриває їх лише тоді, коли користувач розташує свою мітку на запитувачеві, а відомості про користувача порівнюються з тими, що зберігаються у базі даних. Радіо частотна ідентифікація контролює відкриття і закриття дверей. Залежно від джерела електричної енергії, мітки RFID класифікуються як активні або пасивні. Активні мітки вживають батарею для живлення і передачі інформації мітки за запитом запитувача. Але ці теги надзвичайно дорогі й зрідка вживаються. З іншого боку, пасивні мітки отримують енергію від запитувача для живлення їх схеми. Ці мітки надзвичайно вигідні та, як наслідок цього, велика частина додатків вживають їх. Головними перевагами пасивної технології є низька ціна і маленькі габарити. Пасивна мітка RFID передає дані запитувачу, коли вона потрапляє в електромагнітне поле, яке генерується запитувачем. Явище ґрунтоване на законі електромагнітної індукції Фарадея. Струм, який протікає через котушку зчитувача, робить магнітне поле, що сполучається з котушкою трансбордера, тим створюючи струм в котушці трансбордера. Потім котушка трансбордера змінює цей струм, змінюючи навантаження на свою антену. Ця зміна практично є модульованим сигналом, який приймається котушкою зчитувача засобом обопільної індукції між котушками. Котушка зчитувача декодує цей сигнал і передає його на комп'ютер для майбутньої обробки. Додаткове під'єднання антени дозволяє зменшити розміри пристрою.

Є різні типи систематизації міток. Великий популярна і поширена розділяє мітки на чіпові(Рисунок 1.5) і біс чіпові (без мікро схемні), зображені на рисунку 1.6.

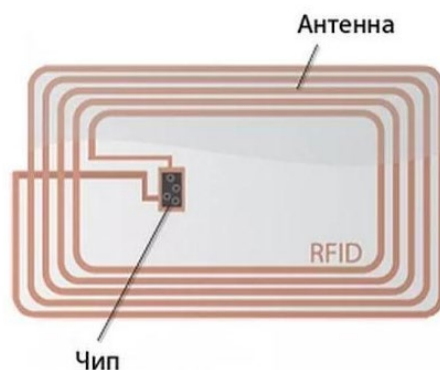


Рисунок 1.5 - Чопова RFID- мітка

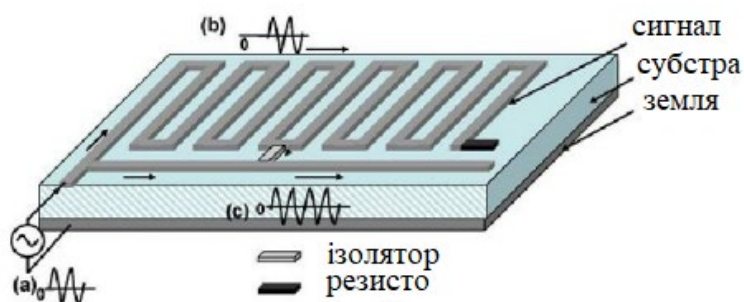


Рисунок 1.6 - Без чопової RFID- мітка

Чіпові мітки містять інтегральну мікросхему — чіп, а без чіпових — ні. Друга систематизація ділить типи міток на пасивні, напівактивні й активні. До складу пасивних міток не заходить інтегроване джерело живлення та активний передавач; напівактивні мітки містять елемент живлення, але не мають активного передавача; активні мітки містять обидва елементи. У черговій систематизації мітки розділяються на лише прочитуванні (read only) і причитуванні/записувач (read/write). Лише причитуванні мітки отримують власний ідентифікаційний код на виробництві. Пам'ять цього типу міток або лише читана (ROM) або одноразово програмована і неодноразово читана (WORM), тобто в процесі роботи з WORM пам'яттю, ідентифікаційний код цього типу міток можна перезаписати один раз. Причитуванні/що записує мітки можуть неодноразово пере програмуватися в процесі використання, і на них можна записати додаткову дані, але не лише серійний номер або код.

По робочій частоті мітки діляться на мітки низькочастотного спектра LF (125-134 кГц), високочастотного спектра HF (13,56 МГц), і надвисокочастотного спектра UHF (860-960 МГц).

2. Проектний розділ

2.1 Розробка електричної схеми і вибір необхідних компонентів

В якості бази, яка триматиме під контролем усю роботу пристрою, був обраний мікроконтролер серії Amica NodeMcu V2 Lua, лицьова частина якого представлена на рисунку 2.1.

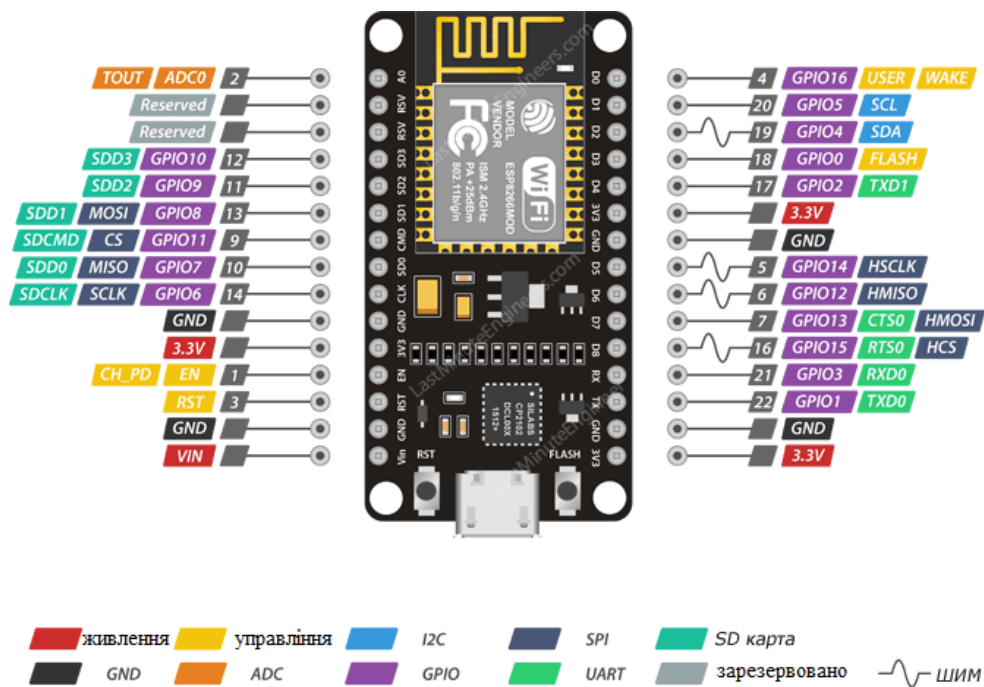


Рисунок 2.1 - Лицьова сторона Amica NodeMcu V2 Lua

Amica NodeMcu V2 Lua - є платою розробника на базі чіпа ESP8266(версія ESP12E), який є UART, - WiFi модуль з ультра низьким споживанням. На платі передбачені дві кнопки, перша підписана як RST і розташована у верхньому лівому кутку, є кнопкою скидання, яка використовується для скидання мікросхеми ESP8266, виведенням RST. Друга кнопка FLASH в лівому нижньому кутку, кнопка завантаження, використовується при оновленні прошивки, виведення GPIO0.

NodeMcu V2 Lua містить в цілому 30 виходів, виведених з обох боків плати, з яких 17 виходів GPIO(контакти модуля ESP - 12E).

Призначення контактів :

1. Виведення VIN - виведення для підключення зовнішнього джерела живлення.

2. Виведення 3V3 і GND - вихід з внутрішнього стабілізатора напруги, можна використати для подання живлення на зовнішні пристрої.

3. RST - виведення RST використовується для скидання мікросхеми ESP8266.

4. EN - коли на виведенні висока напруга мікросхема ESP8266 включена, коли низьке в режимі енергозбереження.

5. CLK(GPIO6), SDO(GPIO7), CMD(GPIO11), SD1(GPIO8), SD2(GPIO9) і SD1(GPIO10) - виведення підключені до Flash чіпу в модулі ESP - 12E, не рекомендується використати.

6. PSW - зарезервовані контакти

7. A0 - виведення вбудованого 10-розрядного АЦП.

8. TX(GPIO1), RX(GPIO3) - контакти для завантаження прошивки і зв'язку з комп'ютером, не рекомендується використати.

9. D0(GPIO16), D1(GPIO5), D2(GPIO4), D3(GPIO0), D4(GPIO2), D5(GPIO14), D6(GPIO15), D7(GPIO13), D8(GPIO15) - виведення для підключення зовнішніх пристроїв.

Так само, деякі виведення GPIO мають додаткові функції :

1. Інтерфейс SPI GPIO12: MOSI, GPIO13: MISO, GPIO14: SCLK, GPIO15: CS

2. Інтерфейс I2C (оскільки ESP8266 не має апаратних виходів I2C, але можна реалізувати програмно). Зазвичай як контакти I2C використовуються наступні виведення GPIO5: SCL, GPIO4: SDA

3. ШИМ(PWM), ESP8266 підтримує програмний ШИМ на усіх виходах введення / виведення : від GPIO0 до GPIO16 з 10-бітовому дозволі.

Для початку роботи треба подати живлення від AC/DC- адаптера або батарейки, або через USB- кабель від персонального комп'ютера.

Основні характеристики Amica NodeMcu V2 Lua представлені нижче

Робоча напруга живлення 5В, рекомендована напруга живлення від 4.5 до 9В, гранична напруга живлення 10В. Містить в цілому 30 виходів з яких 17 виходів GPIO, струм одного виведення дорівнює 15 мА, максимальний струм

виведення 3.3 В і струмом до 600 мА. У мікроконтролера є Flash- пам'ять на 4 МВ, використовується при створенні програм, підтримка WiFi стандарту 802.11 b / g / n, підтримка STA / AP / STA + AP режимів, вбудований стек протоколів TCP / IP з підтримкою множинних клієнтських підключень(до 5), швидкість передачі : 110 - 460800 байтів в секунду, пере прошивка з хмари або через USB, відстань між контактними пінами: 23 мм, розміри плати : 48x26 мм, діапазон робочих температур : - 40 - +125 градусів, маса: 18 г.

В цілях виконання радіочастотної ідентифікації був обраний RFID модуль RC522.Його характеристики представлені нижче.

Напруга живлення 3,3 В, споживаний струм не більше 30 мА, робоча смуга частот 13,55-13,57 МГц, прочитується на відстані 0-25 мм, фізичний розмір зчитувача 40 x 60 мм, робоча температура від 20 до 80С.

Супроводжувані карти: класи S50, S70, Ultralight, Pro, DESFire; типи Mifare S50, Mifare S70, Mifare UltraLight, Mifare Pro, Mifare DESfire. Швидкість передачі інформації 106, 212, 424, 848 кбіт/с. Шифрування Security Features Mifare classic™ (термін Mifare може використати тільки компанія NXP Semiconductors, а також компанії, що мають ліцензію від NXP на виробництво чіпів). Мітки MiFare Classic працюють на високочастотних радіохвилях, зокрема на частоті 13,56 МГц. Це та ж частота, на якій працюють пристрої з підтримкою Near Field Communication(NFC).

У RFID- мітках відсутній мікропроцесор і захищений елемент здатний до аутентифікації. RFID- мітки MiFare були введені NXP Semiconductors в 1995 році, і відтоді був проданий більше мільярда міток у всьому світі. Діючи в якості систем контролю доступу і електронних гаманців, мітки притягнули увагу дослідницьких груп, які провели численні дослідження, що стосуються безпеки, які пропонують мітки. MiFare Classic реалізують власний криптографічний алгоритм під назвою CRYPTO - 1. Це потоковий шифр з 48-бітовим секретним ключем, який використовується для забезпечення конфіденційності даних і взаємної аутентифікації між міткою і зчитувачем.

Зовнішня схема пристрою для читання RFID- міток представлена на

рисунку 2.2.

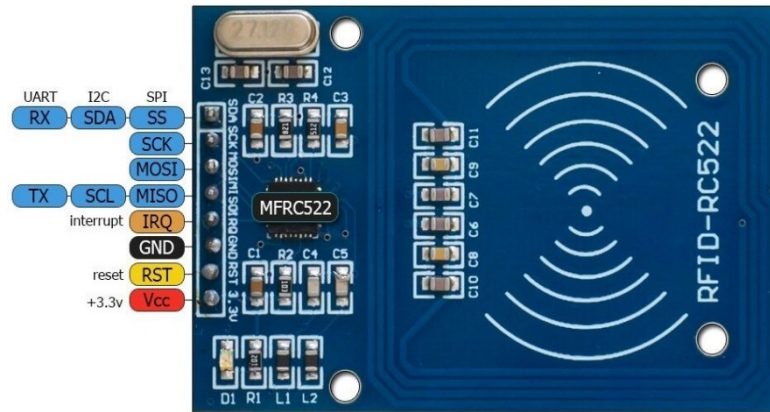


Рисунок 2.2 - Пристрій для читання RC522

Контакти і сигнали RFID RC522:

- VCC - Живлення 3.3V;
- RST - Reset. Лінія скидання, вхід;
- GND - Ground. Земля;
- MISO - Master Input Slave Output - дані від веденого ведучому, вихід SPI;
- MOSI - Master Output Slave Input - дані від ведучого до веденого, вхід SPI;
- SCK - Serial Clock - тактовий сигнал, вхід SPI;
- SDA - Slave Select - вибір веденого, вхід SPI;
- IRQ - лінія переривань, вихід.

зчитувач підтримує інтерфейси SPI, UART і I2C через які відбувається обмін даними з іншими пристроями. На платі модуля RFID RC522 установкою логічних рівнів на особливим виведенням мікросхеми обраний інтерфейс SPI. З одним NodeMcu можуть працювати декілька приладів, підключених до шини SPI.

Підключення модуля RC533 до NodeMcu робиться відповідно до таблиці 2.1. Для нормальної роботи представленого вище модуля вихід IRQ не підключається до NodeMcu.

Таблиця 2.1 - Підключення RC522 до NodeMcu

MFRC522	NodeMcu
RST	D4
SDA	D8

MOSI	D7
MISO	D6
SCK	D5
3.3V	3.3V
GND	GND

У комплекті з цим модулем входить біла пластмасова карта Mifare Classic 1K і мітка у вигляді брелка, зображена на рисунку 2.3.



Рисунок 2.3 - RFID- мітка Mifare 1K

У ній знаходяться антена і мікросхема Mifare S50, яка містить пам'ять і радіо частину. Розмір пам'яті 1 кб, тип EEPROM. Вона розбита на 16 секторів, які складаються з 4 розділів. У кожному розділі три інформаційні частини і одна для ключів. У середині однієї частини є 16 б пам'яті. Термін зберігання даних десять років, циклів перезапису 100000

Неповторність картки Mifare забезпечується привласненням витівником номера, який використовується як ідентифікаційний код. Для захисту даних в мікросхемі карти застосовано апаратне кодування. Під час роботи дані з картки поступають на зчитувач лише після обов'язкової ідентифікації коду, що записаного в сектор пам'яті картки і лежить в зчитувачі.

Для роботи в середовищі розробки Arduino можна користуватися різними

сторонніми бібліотеками, які були розроблені з тією метою, щоб значно полегшити роботу зі зчитувачем RC522. Для запису і читання з карти слід знати її неповторний номер, потрібний для роботи системи радіочастотної ідентифікації. З цією метою необхідно завантажити програмку з переліку прикладів бібліотеки RFID під заголовком "CardInfo", підключити RC522 до NodeMcu, і запустити програмку в середовищі розробки Arduino IDE. При знаходженні робочої мітки в зоні діяння RFID - зчитувача на моніторі порту покажеться інформація про карту, яка була представлена на рисунку 2.4.

```
Card found
Cardnumber:
Dec: 85 , 150 , 29 , 212 , 46
Hex: 55 , 96 , 1D , D4 , 2E
```

Рисунок 2.4 - Ідентифікаційні номери RFID – міток

Програма виводить ряд чисел: 85, 150, 29, 212, 46. Треба записати їх в зворотному порядку. перша кількість виключається(контрольна сума), яке спочатку було останнім, а числа, що залишилися, переводяться в шістнадцятиричний код. Потім вони записуються в тому ж порядку, але без пропусків. Придбане величезне число треба перевести в десятковий код, внаслідок чого вийде ідентифікаційний номер карти. З його допомогою вже можна проводити різні маніпуляції і складати різні програми, приміром системи контролю доступу в приміщення.

У облаштуванні майже усіх електронних замків може знаходитися елемент, що відтворює звук. Для даних задач підійде зумер випромінювач звуку, який може відтворити звук на базі п'єзоелектричного ефекту. П'єз динамік, зображений на рисунку 2.5, складається із залізної пластинки, з нанесеної на ній п'єзоелектричної кераміки, яка має струмопровідне напилення. Пластинка і напилення є контактами п'єз випромінювача, полярність яких плюс і мінус. Якщо до контактів прикласти напругу, під дією оборотного п'єзоелектричного ефекту випромінювач почне відтворювати звук, а якщо механічно вплинути на п'єз елемент, то на його контактах покажеться напруга. Щоб під'єднати п'єз динамік до мікроконтролера, контакт, позначений знаком "+" підключається до хоч якого

цифрового входу, а мінусовий контакт до виходу GND.



Рисунок 2.5 - Пьезокерамічний сигналізатор SFM - 27 95 дБ 3-24 В

Для комутації різних пристроїв вживається особливий одно каналний модуль реле для мікроконтролерів, зображений на рисунку 2.6, а принципова схема пристрою - на рисунку 2.7.



Рисунок 2.6 - Схемне зображення реле(вигляд зверху)

До складу реле входять: резистори номіналом 1 кОм(R1, R2), який підтягує резистор R3 на 10 кОм, р - н - р транзистор(VT1), оборотний діод (VD2) і реле(K1). VD1(червоний світлодіод) - індикація подання живлення на модуль, загоряння VD3(зелений світлодіод) говорить про замикання реле. Контакти реле показані на рисунку 2.8.

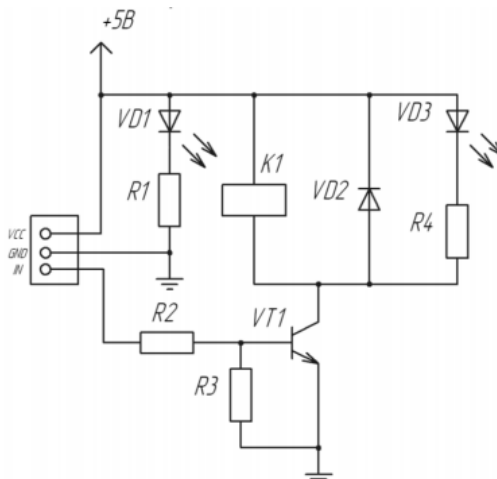


Рисунок 2.7 - Принципова схема модуля реле

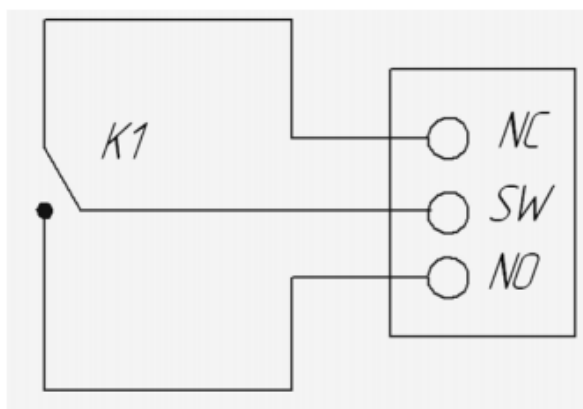


Рисунок 2.8 - Контакти одно каналного реле

При включенні виведення знаходяться у високому стані, транзистор не відкритий. Тому що транзистор р - n - р типу, то для його відкриття необхідно подати на базу мінус. З цією метою необхідно застосовувати функцію `digital Write(pin, LOW)`. Реле спрацьовує, коли транзистор відкритий і через ланцюг, що управляє, тече струм. Для відключення реле слід закрити транзистор, подавши на базу плюс за допомогою функції `digital Write(pin, HIGH)`. Контакти реле NC - нормально замкнутий, SW - контакт перемикавання, NO - нормально розімкнений.

Для швидкого та зручного відкриття із середини використовуються механічна кнопка призначена для передачі електричного сигналу різних пристроїв методом замикання або розмикання 2-ох або більше контактів. Кнопка - DS - 213 OFF -(ON) показана на малюнках 2.9-2.10.



Рисунок 2.9.Кнопка - DS - 213 OFF -(ON).

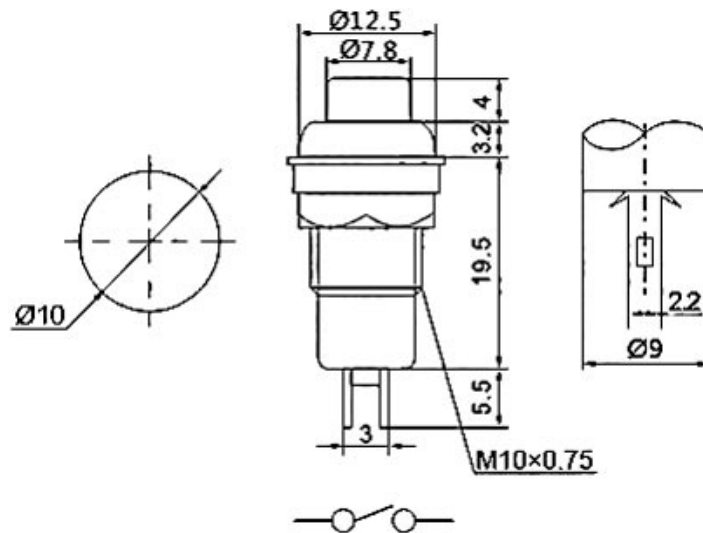


Рисунок 2.10.Схема кнопки - DS - 213 OFF -(ON).

Кнопка - DS - 213 OFF -(ON) має подальші властивості:

- Функція OFF -(ON)
- Максимальна робоча напруга 250V
- Максимальний робочий струм 1A
- Рівень захисту IP40
- Діапазон робочих температур - 25°C~+85°C
- Опір контактів <20mΩ Опір ізоляції >500VDC 100MΩ Minute
- Напрацювання на відмову(механічна) >50000 циклів
- Напрацювання на відмову(електрична) >10000 циклів
- Сертифікат CE, ROHS

В якості замикаючого пристрою вибраний Електромеханічний соленоїдний замок 12 V. У разі потреби можлива швидка зміна на інший замикальний пристрій. Електромеханічний соленоїдний замок 12 V показаний на рисунку 2.11.

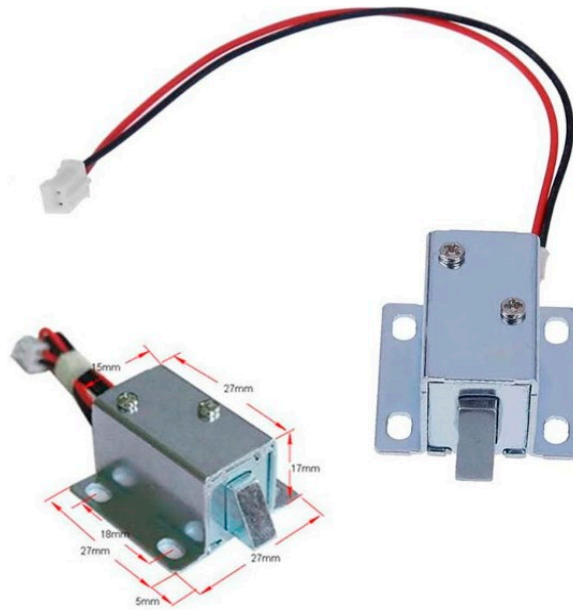


Рисунок 2.11. Електромеханічний соленоїдний замок 12 V.

Основні характеристики:

- Живлення: 12 В
- Довжина засувки : 13 мм
- Розмір(не включаючи засувку) : 5.5 x 4.2 x 2.8 см
- Вага: 130 г

2.2 Розробка конструкції пристрою

Для представлення конструкції облаштування контролю доступу скористаємося надзвичайно комфортною програмою для візуалізації усіх кроків роботи з NodeMCU. За допомогою програми Fritzing beta з електронного ресурсу fritzing.org, яка поширюється на безкоштовній базі, можна моделювати різні рішення, які будуть показані в комфортній для користувача формі макету або схеми підключення.

Промодельємо модулі за допомогою Fritzing beta, які використовуються разом з NodeMCU, і виведемо їх схеми підключення до NodeMCU V2 Lua, які будуть показані на малюнках 2.12 - 2.16.

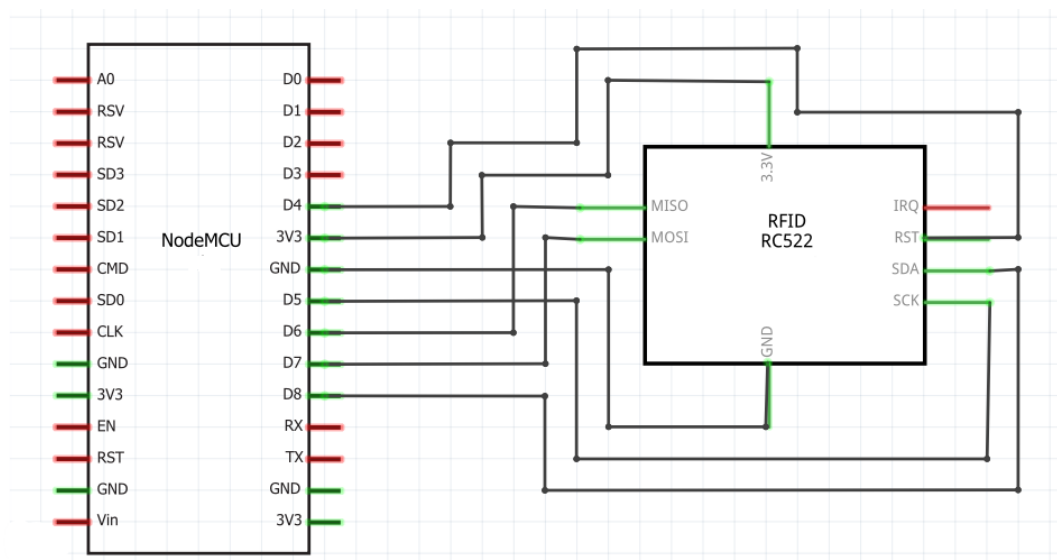


Рисунок 2.12 - Схема підключення RFID-зчитувача RC522

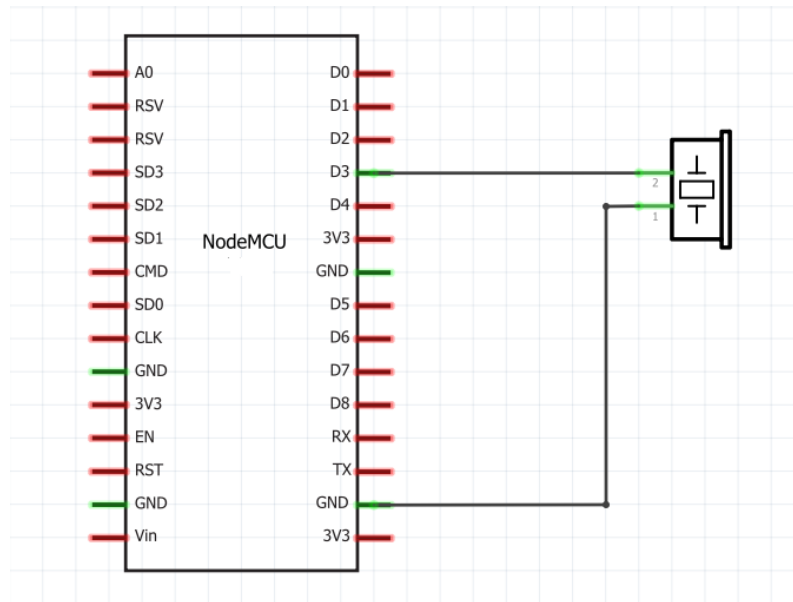


Рисунок 2.13 - Схема підключення пьезокерамічного сигналізатор

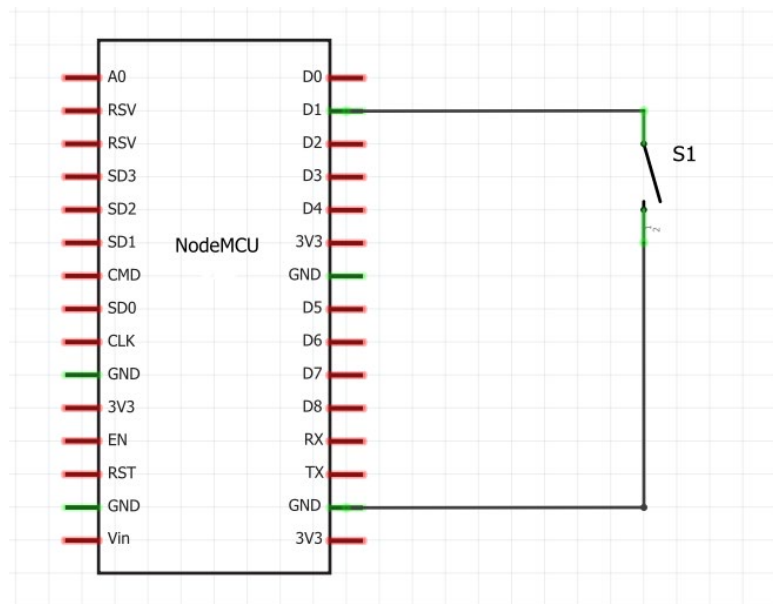


Рисунок 2.14 - Схема підключення кнопки - DS – 213

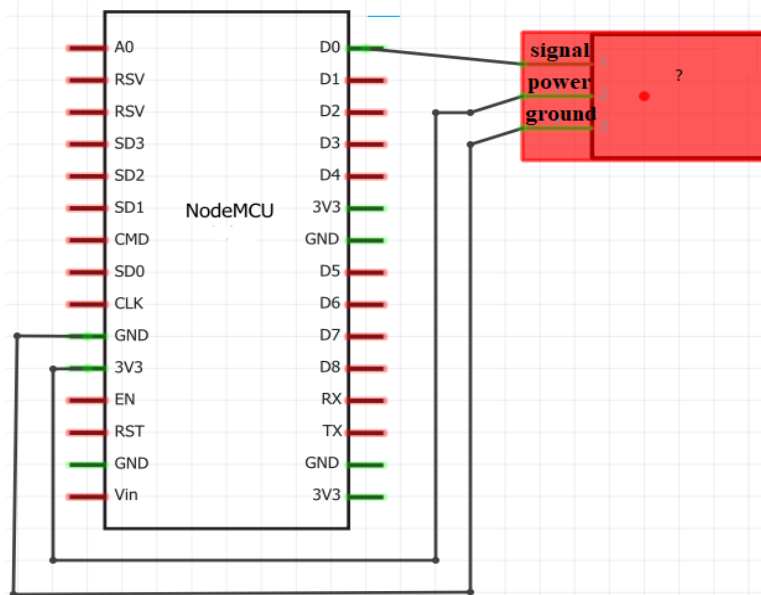


Рисунок 2.15 - Схема підключення одно каналного реле.

У результаті поєднали усі модулі на одній схемі підключення, представлений на рисунку 2.16

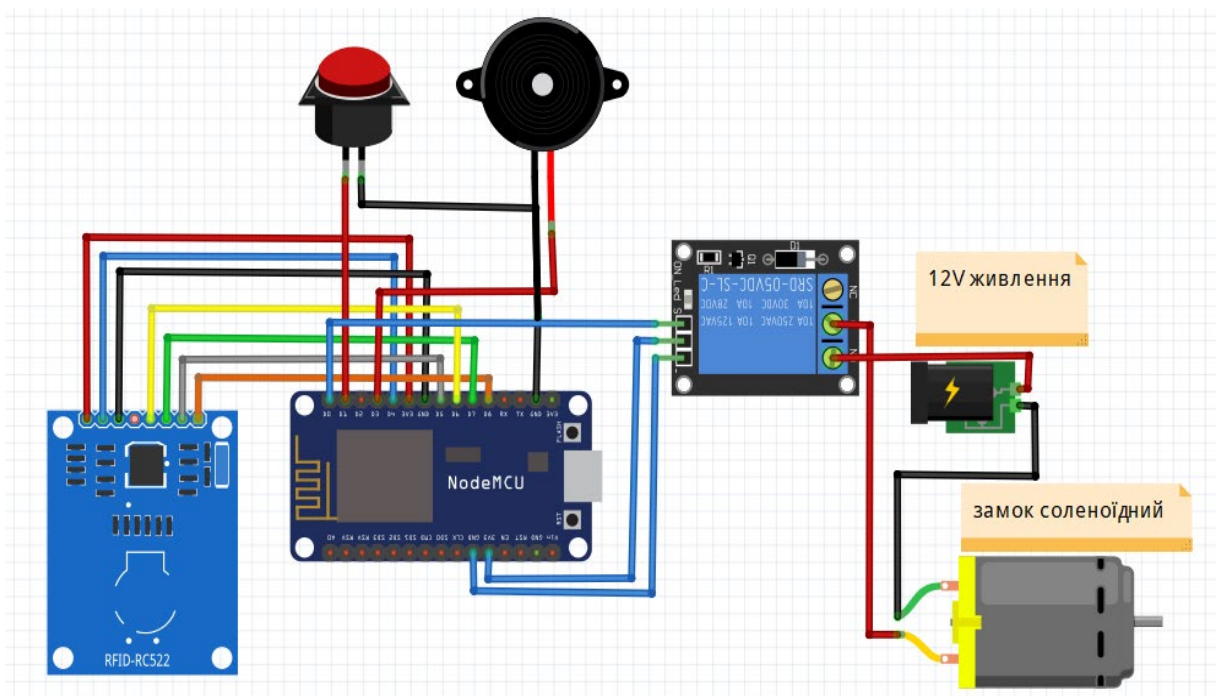


Рисунок 2.16 - Схема підключення

2.3 Розробка програмної частини пристрою

Для того щоб почати розробляти програму треба спочатку задатися базисним методом роботи пристрою, потрібним для усвідомлення дій виконання різних ситуацій(Рисунок 2.17).

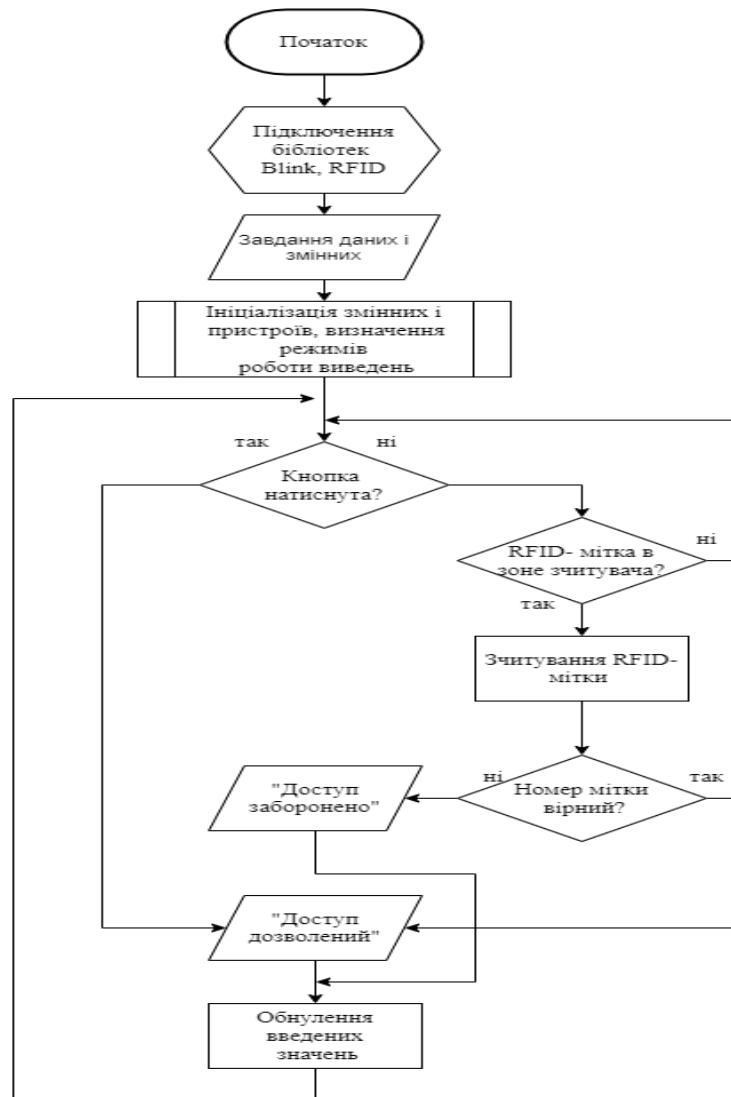


Рисунок 2.17 - Базовий алгоритм роботи облаштування контролю доступу.

Для написання програми скористаємося спеціально розробленим середовищем розробки Arduino IDE, яка поширюється у вільному доступі в мережі Веб і яку можна знайти на інтернет представництві розробника www.arduino.cc та додатком Blynk - це додаток який дозволяє управляти пристроями через мережу веб, за допомогою хмари та конструювати і підключати інтерфейс додатку до пристроїв.

Arduino IDE - інтегроване середовище розробки для Вiндовс, MacOS i Лiнукс, яка була розроблена на Сi i С ++, створена для створення i завантаження програм на Arduino- сумiснi плати, також на плати iнших виробникiв.

iнтерфейс Arduino IDE порiвняно простий в освоєннi, його основою є С++ схожа мова програмування iз зумовленими функцiями. Для програмування Arduino використовується полегшена версiя мови С++. Як i в iнших Си- схожих мовах програмування є ряд правил написання коду. Так само як i С++ мова є такою, що жорстко типiзується i компiльованим.

Blynk це служба веб речей (IoT), розроблена для того, щоб зробити дистанцiйне керування i причитування цих датчикiв з ваших пристроїв максимально швидким i простим.

У платформi є три основнi компоненти:

- Blynk Server - вiдповiдальний за всi комунiкацiї мiж смартфоном i апаратним забезпеченням.
- Blynk Libraries - для всiх популярних апаратних платформ - дозволяє спiлкування з сервером i обробляти всi вхiднi та вихiднi команди.
- Blynk App - дозволяє створювати iнтерфейси для ваших проектiв за допомогою рiзних вiджетiв.

Особливостi:

- Подiбний API & UI для Всього пiдтримуваного обладнання та пристроїв
- Пiдключення до хмари використовується:
- GSM
- Ethernet
- USB (Serial)
- WiFi
- Bluetooth and BLE
- Набiр простих у використаннi вiджетiв
- Прямий манiпуляцiї з пiнами без написання коду

- Легко інтегрувати та додавати нові функції за допомогою віртуальних контактів
- Історія моніторингу даних через віджет History Graph
- Зв'язок між пристроями за допомогою Bridge widget
- Надсилання електронних листів, твітів, push-сповіщень тощо.

Для використання Blynk потрібно пройти реєстрація .Проведемо огляд додатку.

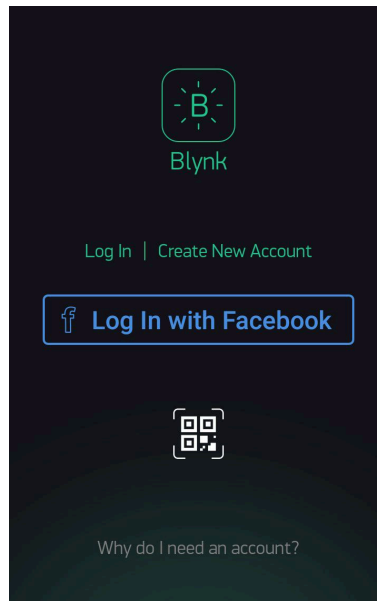


Рисунок 2.18 – Перша сторінка.

На першій сторінці ми можемо:

- Увійти
- Створити новий обліковий запис
- Увійти за допомогою Facebook
- Відскакувати QR-код
- Та прочитати навіщо потрібен акаунт.

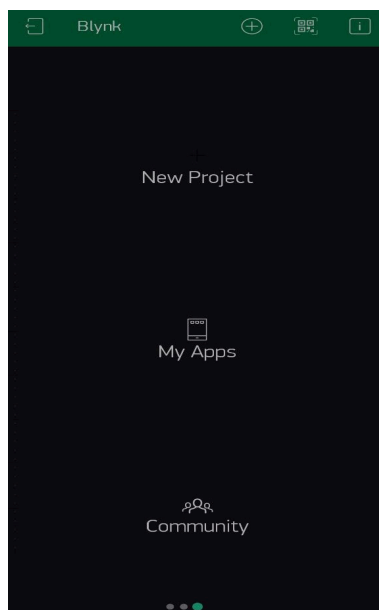


Рисунок 2.19 – Друга сторінка яка є головною.

На головній сторінці ми можемо :

- створити проект
- завантажити вже створений проект
- та зайти до спільноти людей

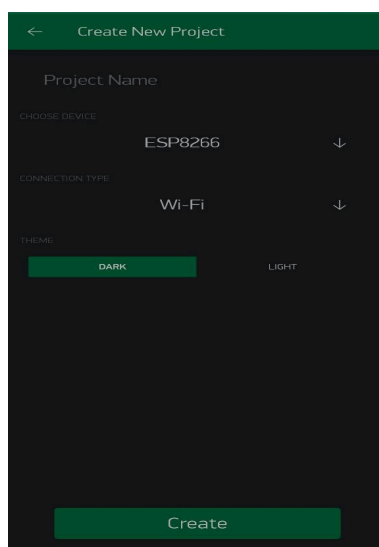


Рисунок 2.20 – сторінка створення проекту.

На сторінці створення проекту ми можемо:

- дати назву проекту
- вибрати мікроконтролер на базі якого будемо будувати проект
- вибрати канал зв'язку через який будемо керувати пристроєм.

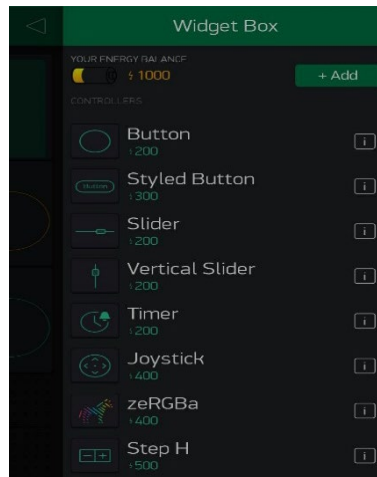


Рисунок 2.21 Панель віджетів.

На панелі віджетів ми бачимо перелік віджетів вони мають пояснення та інформацію щодо підключення та мають фрагменти коду для прикладу.

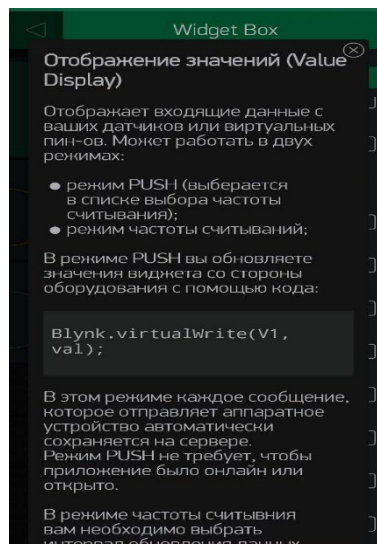


Рисунок 2.22 Довідка віджиту.

Зрештою, після довгого процесу розробки методом проб і прорахунків була складена програма, за допомогою якої можна утілити основний функціонал роботи електричного замка на базі NodeMCU V2 Lua. Для правильної роботи програми треба завчасно встановити декілька сторонніх бібліотек або через програмку "Arduino" - "Скетч" - "Приєднати бібліотеку" - "Управляти бібліотеками" - "Пошук" (рисунки 2.23, 2.24), або через ресурси мережі Інтернет, встановивши її по дорозі "Arduino" - "Скетч" - "Приєднати бібліотеку" - "Додати.ZIP бібліотеку" (рисунки 2.25, 2.26). Потрібні сторонні бібліотеки для роботи з програмою: "ESP8266WiFi.h", "BlynkSimpleEsp8266.h", "MFRC522", "SPI.h", "EEPROM.h".

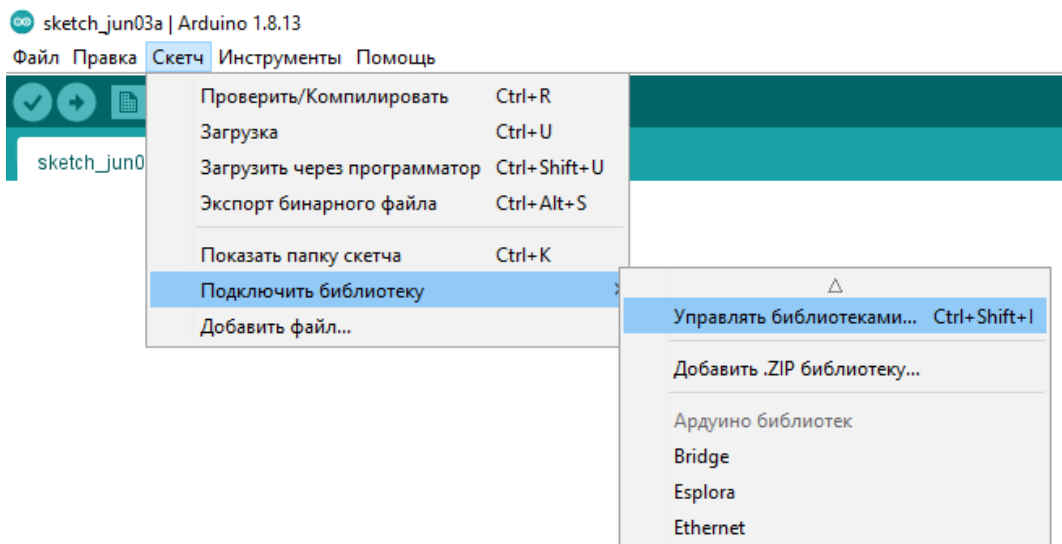


Рисунок 2.23 – Перехід в управління бібліотек.

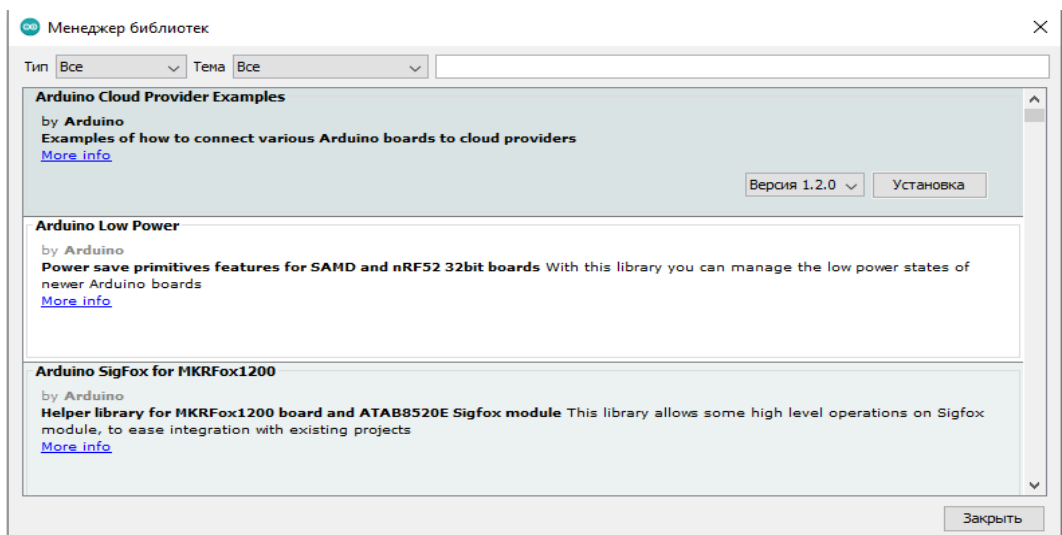


Рисунок 2.24 – Менеджер бібліотек.

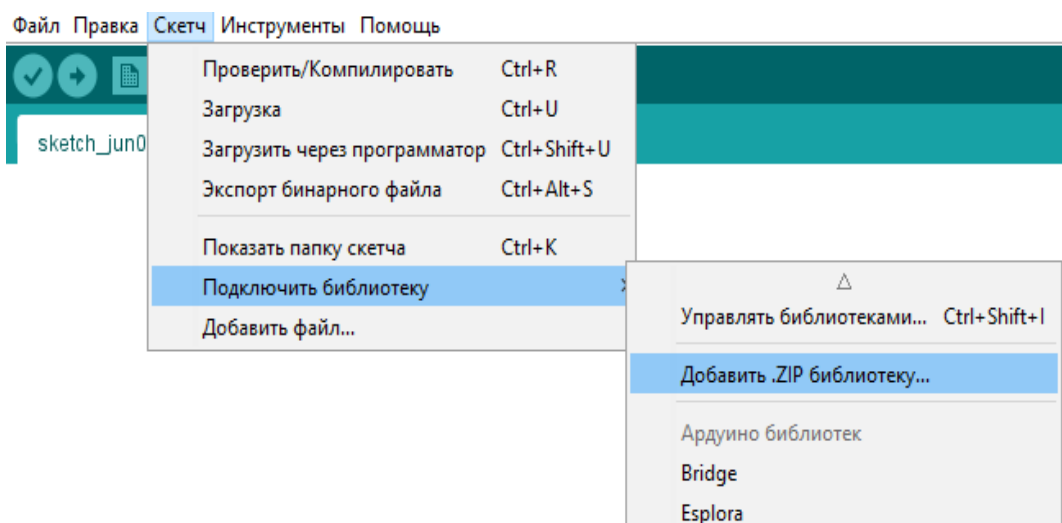


Рисунок 2.25 – Перехід в добавлення ZIP бібліотек.

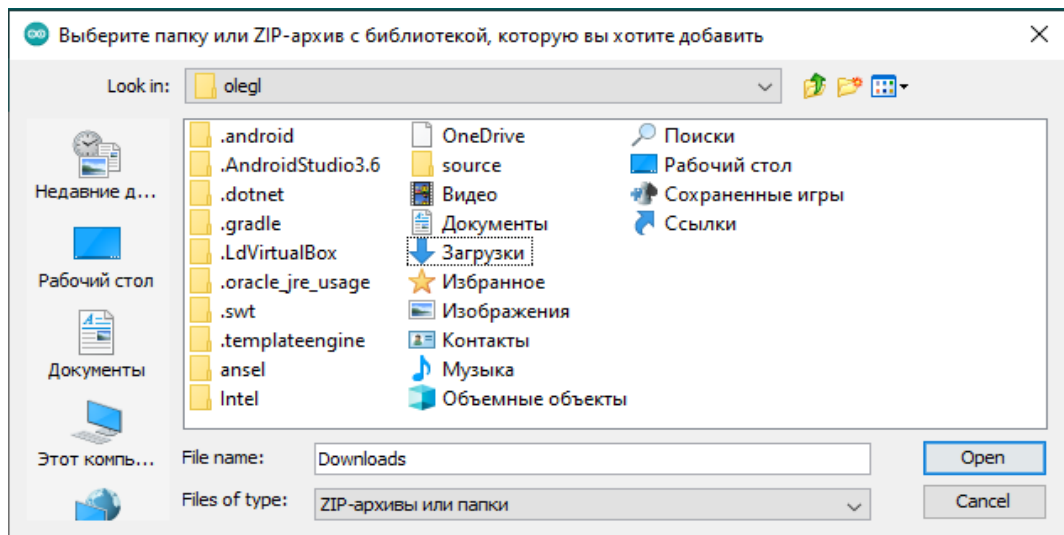


Рисунок 2.26 – Вікно для пошуку ZIP бібліотек.

Програма, яка реалізує потенціал облаштування контролю доступу в приміщення приведена в додатку А.

3. Практична реалізація проекту

3.1. Виготовлення системи

Для підключення різних модулів до мікроконтролера NodeMCU існує декілька способів:

1) використати спеціальну макетну плату, на якій зручно розташовувати різні елементи схеми, а також сполучати їх дротами між собою і мікроконтролером. Яку можна побачити на рисунку 3.1

2) Використати дроти мама-мама якими можна зручніше з'єднати елементи між собою оскільки вони займають менше місця і мають гнучкішу структуру що дозволяє зручно розташувати їх усередині корпусу які можна побачити на рисунку 3.2

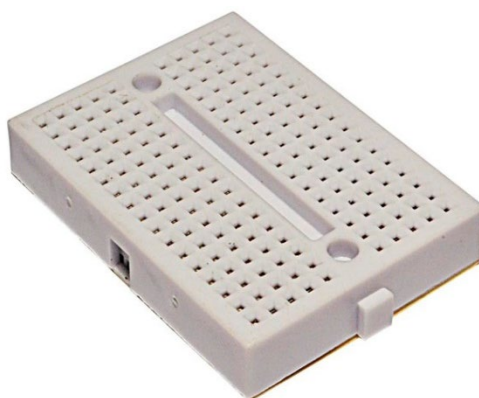


Рисунок 3.1-макетна плата



Рисунок 3.2- дроти мама-мама

Було вибрано з'єднання дротами мама-мама зрештою, вийшла схема, представлена на рисунках 3.3. та 3.4. Усі елементи сполучені з мікроконтролером згідно з принциповими схемами, які були розроблені в спеціалізованій програмі

моделювання Fritzing. і розміщені в картонній коробці.



Рисунок 3.3- вигляд зверху.

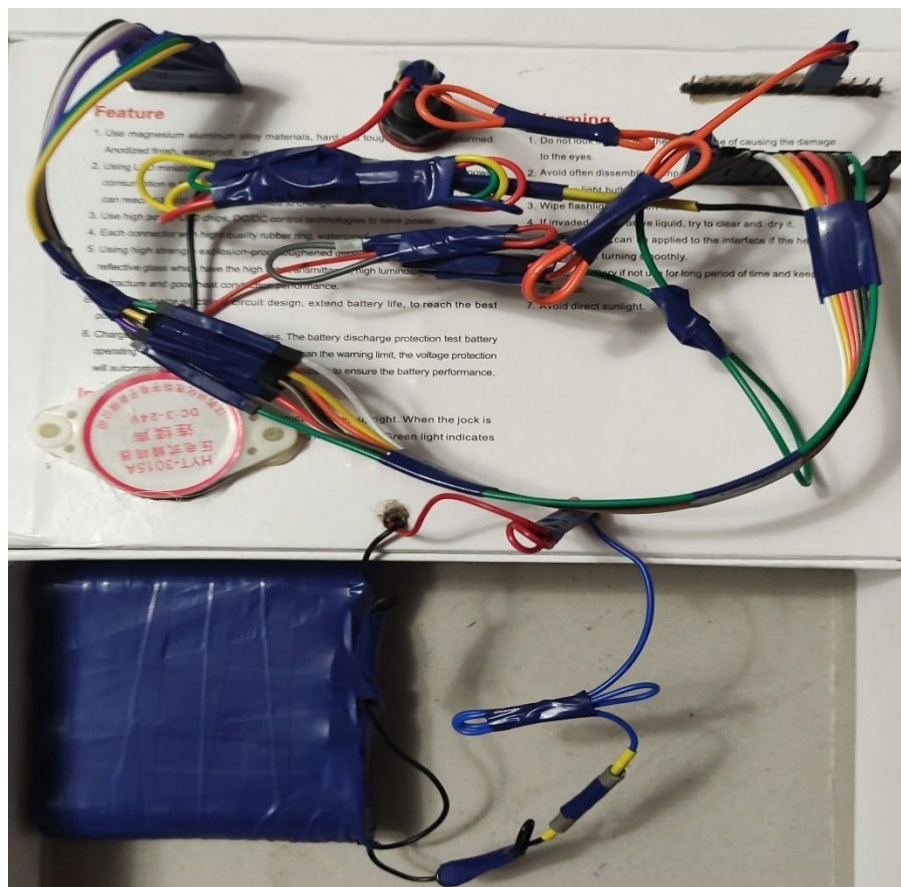


Рисунок 3.4- вигляд усередині.

3.2. Перевірка і відладка програмної частини пристрою

В процесі практичної роботи та створення моделі електричного облаштування контролю доступу, з'явилися деякі труднощі, також нові думки по удосконаленню програмної частини. А саме, була зроблена підміна робочої бібліотеки RFID з Rfid.h на MFRC522, у якої функціонал перевершував попередню бібліотеку. У результаті, найосновнішим рішенням було додавання програмної роботи з енергонезалежною пере записуваною пам'яттю EEPROM мікроконтролера, внаслідок чого ідентифікаційні номери RFID - міток. Вони стали записуватися в цю енергонезалежну пам'ять і при роботі програми виводитися в послідовний порт комп'ютера, а не задаватися в самій програмі, як було до цього.

Разом з EEPROM було додано спеціальні віртуальні кнопки у додатку Blynk також для більш зручного відстеження інформації було добавлено віртуальний дисплей який працює через віртуальний пін та додаток Blynk.

Було добавлено дві кнопки для керування EEPROM пам'яттю та кнопка для відкриття замка через телефон, додаток має наступний вигляд рисунок 3.5.

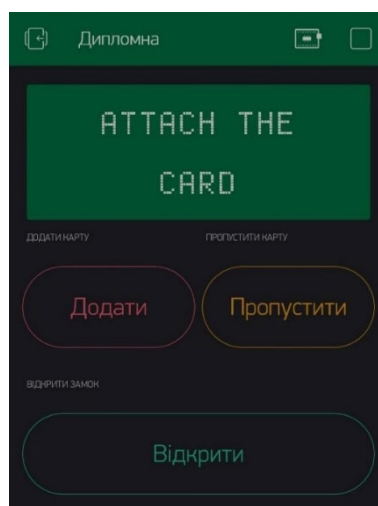


Рисунок 3.5 – Вигляд роботи у мобільному додатку Blynk

Перша кнопка відповідає за перехід у меню управління та додавання нової RFID – мітки або перезапис у пам'яті, після запису однієї комірки пам'яті переходить автоматично на наступну, підключення можливо побачити на рисунках 3.5-3.6.

```

.
// Віджет кнопки на вивід V1
BLYNK_WRITE(V1) {
  int a = param.asInt();
  if (a == 1) beginCard = 1; else beginCard = 0;
}

```

Рисунок 3.5 – Підключення та налаштування першої кнопки у кодї.

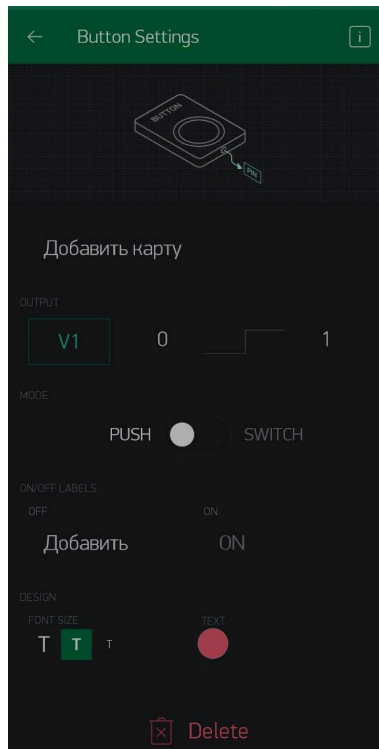


Рисунок 3.6 – Підключення та налаштування першої кнопки у додатку Blynk.

Друга кнопка відповідає за перехід у меню управління та дає можливість перейти на наступну комірку пам'яті без зміни даних у попередній підключення показано на рисунках 3.7-3.8.

```

// Віджет кнопки на вивід V2
BLYNK_WRITE(V2) {
  int a = param.asInt();
  if (a == 1) {
    skipCard = 1;
    if (EEPROMstartAddr / 5 < 10) EEPROMwriteUIDcard();
  } else {
    skipCard = 0;
  }
}

```

Рисунок 3.7 – Підключення та налаштування другої кнопки у кодї.

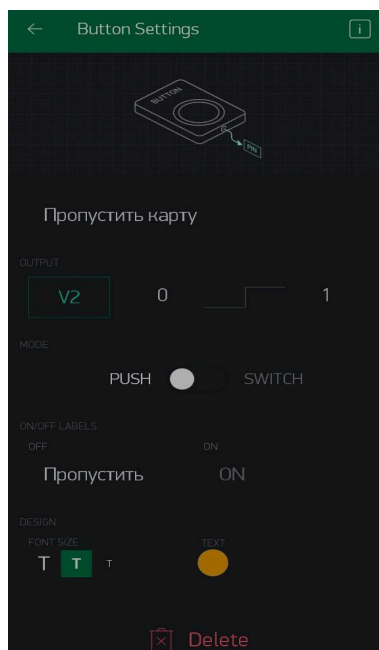


Рисунок 3.8 – Підключення та налаштування другої кнопки у додатку Blynk.

Третя кнопка дозволяє відчинити замок підключення та налаштування на рисунках 3.9-3.10.

```
// Віджет кнопки на вивід V3
BLYNK_WRITE(V3) {
  int a = param.asInt();
  if (a == 1) {
    digitalWrite(SLN_PIN, LOW); //Відкрито
    lcd.print(0, 0, " APP UNLOCK OK "); // Вивести на дисплей
    lcd.print(0, 1, " DOOR OPEN "); // Вивести на дисплей
    digitalWrite(PiezoPin, HIGH), delay(200), digitalWrite(PiezoPin, LOW); // Подати звуковий сигнал
    delay(2000); // зачекати 2000 Мілі секунд
    DisplayWait_CARD();
  }
}
```

Рисунок 3.9 – Підключення та налаштування третьої кнопки у кодї.

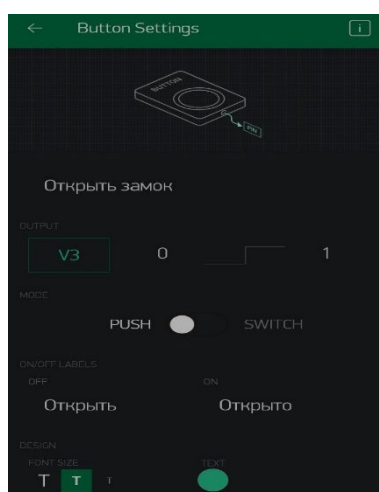


Рисунок 3.10 – Підключення та налаштування третьої кнопки у додатку Blynk.

Дисплей дозволяє виводити інформацію на англійській мові та латинськими цифрами. Спосіб підключення та код програми видно на рисунках 3.11-12.

`WidgetLCD lcd(V0);` // ініціалізація дисплея на віртуальний пін V0

Рисунок 3.11 – Підключення та налаштування дисплея у коді.

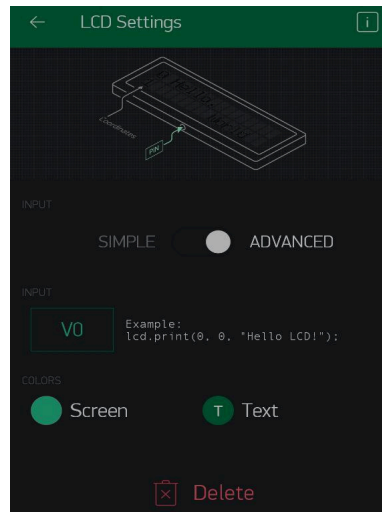


Рисунок 3.12 – Підключення та налаштування дисплея у додатку Blynk.

На нього виводиться інформація про те у якому режимі знаходиться замок якщо жоден із способів відкриття не використовується у даний момент то ми бачимо наступний напис на ньому зображений на рисунку 3.13.



Рисунок 3.13 – Початкове зображення дисплею.

Якщо був використаний спосіб відкриття за допомогою кнопки виводиться наступний текст рисунок 3.14.



Рисунок 3.14 -Зображення на дисплею при відкритті кнопкою.

Якщо за допомогою RFID – мітки бачимо наступний текст де друга строчка це UID карти рисунок 3.15.

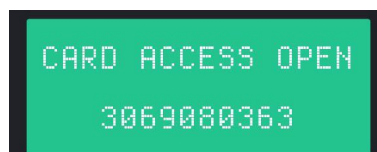
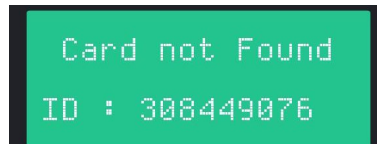


Рисунок 3.15 -Зображення на дисплею при відкритті замка RFID – міткою.

Якщо RFID – мітки не було знайдено як збереженої маємо наступний текст

рисунок 3.16.



```
Card not Found
ID : 308449076
```

Рисунок 3.16 -Зображення на дисплею при спробі відкрити не збереженою RFID – міткою.


При переході в режим управління EEPROM пам'яттю виводиться текст рисунок 3.17.



```
START REC CARD
PLEASE TAG CARDS
```

Рисунок 3.17 - Зображення на дисплею при переході в режим управління EEPROM пам'яттю

Запис до EEPROM пам'яттю RFID – мітки має наступний вигляд на дисплеї де друга строчка це номер комірки рисунок 3.18.



```
RECORD OK! IN
MEMORY : 2
```

Рисунок 3.18 - Зображення на дисплею при збереженні RFID – мітки.

Перехід до наступної комірки EEPROM пам'яті без запису інформації має наступний вигляд на дисплеї де друга строчка це номер комірки рисунок 3.19.



```
SKIP RECORD
label : 4
```

Рисунок 3.19 - Зображення на дисплею при переході на наступну комірку без запису в попередню інформації

Після закінчення редагування EEPROM пам'яті виводиться наступний текст рисунок 3.20.



```
RECORD FINISH
```

Рисунок 3.20 - Зображення на дисплею при закінченні налаштування.

Матеріальні витрати для реалізації проекту

Під час побудови цього проекту, основною задачею було зробити цей проект багатофункціональним і з мінімальними витратами. Далі приведена таблиця з найменуваннями девайсів і їх цінами.

Назва комплектуючого	Ціна грн/штуку
NodeMCU V2 Lua WI-FI на базі ESP8266 и CP2102	95,00 грн
Модуль читання rfid міток MFRC- 522	52,00 грн
Кнопка DS-213 OFF- (ON), ЧЕРВОНА, без фіксації	8,00 грн
Одно каналне реле для Arduino 5VDC	17,00 грн
Звуковий сигналізатор SFM-27 95 дБ 3-24 В	17,00 грн
Електронний замок соленоїдний	150 грн
З'єднувальний провід	16,00 грн
Загальна сума	355 грн

Висновок

У такій випускній кваліфікаційній роботі здійснена розробка, створення робочої програми, виготовлення, відладка і працюючого макету облаштування контролю доступу в приміщення на мікроконтролерном управлінні.

Пристрій моделює головні функції контролю за різними датчиками і пристроями, які входять в периферію електронної системи.

Макет електричного замку дозволяє змодельовати головні функції і завдання по контролю доступу в різні приміщення за допомогою радіочастотної мітки, яка використовується як ключ- карти і програми Blynk . Модуль реле і приєднаний до його контактів Соленоїдний замок симулюють відкрите або закрите стани дверей в приміщення. При втраті ключа передбачена можливість записати нові ключ карти в цілях вказівки нового ключа, використовуваного для відкриття замку.

Уся налагоджувальна інформація поступатиме на комп'ютер лише при підключенні до нього мікроконтроллера через USB- інтерфейс. У результаті, пристрій може працювати автономною через приєднаний мережевий блок живлення, але в цьому випадку стає недоступною налагоджувальна інформація.

Головним завданням цієї роботи було створення макету облаштування контролю доступу на мікроконтролерном управлінні. Придбана система дозволяє змодельовати головні функції, доступні таким же механізмам без мікроконтролерів.

Список використовуваної літератури

1. Початок роботи з ESP8266 NodeMcu v3 Lua з WiFi. [Електронний ресурс]. URL: <https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/>
2. NodeMCU V2 Lua WI-FI на базі ESP8266 и CP2102. [Електронний ресурс]. URL: <https://arduino-geek.pp.ua/product/nodemcu-v3-lua-wi-fi-esp8266-cp2102/>
3. Модуль читання rfid міток MFRC – 522. [Електронний ресурс]. URL: <https://arduino-geek.pp.ua/product/modul-chtenija-rfid-metok-mfrc-522/>
4. Кнопка DS - 213 OFF -(ON), ЧЕРВОНА, без фіксації. [Електронний ресурс]. URL: <https://arduino-geek.pp.ua/product/ds-213-off-on/>
5. Звуковий сигналізатор SFM - 27 95 дБ 3-24 В. [Електронний ресурс]. URL: <https://arduino-geek.pp.ua/product/звуковой-сигнализатор-sfm-27-95-дб-3-24-в/>
6. Одно канальне реле для Arduino 5VDC. [Електронний ресурс]. URL: <https://arduino-geek.pp.ua/product/rele-odnokanalnoe-5vdc/>
7. Дроти 20см Мама-мама [Електронний ресурс]. URL: <https://arduino-geek.pp.ua/product/1000607/>
8. Підключення блоку реле до модуля NodeMCU [Електронний ресурс]. URL: <https://arduino-kit.ru/blogs/blog/64-podklyuchenie-bloka-rele-k-modulyu-nodemcu>
9. Fritzing. [Электронный ресурс]. URL: <http://fritzing.org/home/>
10. Підключення пьезоизлучателя до NodeMCU. [Електронний ресурс]. URL: <https://arduino-kit.ru/blogs/blog/55-svetodiodnaya-indikatsiya-i-zvukovaya-signalizatsiya-o-kriticheskikh-parametrah-datchikov-dlya-nodemcu>
11. A Review on Chipless RFID Tag Design / A.Hashemi, A.H.Sarhaddi, H.Emami // Majlesi Journal of Electrical Engineering, vol.7, No.2, P.68-75, June 2013. [Електронний ресурс]. URL: https://www.researchgate.net/publication/260190506_A_Review_on_Chipless_RFID_Tag_Design

12. RFID Based Security and Access Control System / U.Farooq, Mahmood ul Hasan, M.Amar et al. // IACSIT International Journal of Engineering and Technology, Vol. 6, No.4, P.309-314, August 2014. [Електронний ресурс]. URL: https://www.researchgate.net/publication/275685766_RFID_Based_Security_and_Access_Control_System
13. NodeMCU ESP8266 [Електронний ресурс]. URL: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>
14. Blynk - інтернет пульт для ваших ардуино пристроїв. [Електронний ресурс]. URL: <https://the-robot.ru/study/nodemcu-blynk-uroki-1-ustrojsva-vyvoda/>
15. Блог Программиста [Електронний ресурс]. URL: <https://blog-programmista.ru/post/80-arduino-ide-cto-eto-za-programma.html>
16. Blynk [Електронний ресурс]. URL: <https://shoorik007.github.io/#intro>
17. Різниця між UID і серійним номером карти [Електронний ресурс]. URL: <https://coderoad.ru/17608670/MiFare-разница-между-UID-и-серийным-номером-карты-MiFare>

Додаток А

Лістинг програми облаштування контролю доступу в приміщення

Ініціалізація бібліотек, необхідних для спрощення роботи з різними модулями : ESP8266WiFi.h - бібліотека за допомогою якої ESP8266 здійснює підключення до WiFi; BlynkSimpleEsp8266.h бібліотека використовується для підключення Blynk; SPI.h - бібліотека для роботи з пристроями, підтримувальними SPI протокол; MFRC522.h - бібліотека для роботи з RFID-модулем RC522; EEPROM.h - стандартна бібліотека роботи з енергонезалежною пере записуваною пам'яттю EEPROM у подетсу;

Початок програми :

```
#include <ESP8266WiFi.h>
```

```
#include <BlynkSimpleEsp8266.h>
```

```
#include <SPI.h>
```

```
#include <MFRC522.h>
```

```
#include <EEPROM.h>
```

```
//Визначення основних пінов, до яких підключаються різні модулі:
```

```
#define SS_PIN D8 // RFID SS
```

```
#define RST_PIN D4 // RFID RST
```

```
#define BTN_PIN D1 // Кнопка
```

```
#define SLN_PIN D0 // підключення реле
```

```
//Ініціалізація RFID - зчитувача:
```

```
MFRC522 mfrc522(SS_PIN, RST_PIN); // об'єкт MFRC522
```

```
unsigned long uidDec, uidDecTemp; // для відображення номера картки в  
//десятковому форматі
```

```
int ARRAYindexUIDcard; // Індекс для масиву CardUIDeEPROMread.
```

```
int EEPROMstartAddr; // Стартовий елемент пам'яті для запису / читання  
//EEPROM "Card UID".
```

```
long adminID = 1122334455;
```

```
bool beginCard = 0;
```

```
bool addCard = 1;
```

```
bool skipCard = 0;
```

```
int LockSwitch;
```

```
unsigned long CardUIDeEPROMread[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}; // Масив
```

```
// для зберігання 10 "UID картки" / унікальних номерів карт NFC, зчитаних  
// EEPROM.
```

```
// Пин, к которому подключен пьез динамик.
```

```

int PiezoPin = D3; // Piezo / Динамік.

char auth[] = "tqSS8EvtqQXV0_rSAkY7wEcX_H5RPWMX"; // введіть ваш
// токен авторизації
char ssid[] = "MyNet"; // введіть Ім'я вашої мережі
char pass[] = "79138246"; // введіть свій пароль від мережі

WidgetLCD lcd(V0); // ініціалізація дисплея на віртуальний пін V0

void setup() {
Serial.begin(9600); // ініціалізація послідовного порту
pinMode(SLN_PIN, OUTPUT); // встановлюємо режим роботи виведення, як
// "вихід"
digitalWrite(SLN_PIN, HIGH); // підключення реле
pinMode(BTN_PIN, INPUT_PULLUP); // кнопка для відкриття замку
pinMode(PiezoPin, OUTPUT); // устанавлюємо режим роботи вивода, як
// "ВЫХОД"

SPI.begin (); // ініціалізація SPI
mfrc522.PCD_Init (); // ініціалізація MFRC522

Blynk.begin(auth, ssid, pass);
lcd.clear(); // Використовується, щоб очистити віджет LCD
EEPROM.begin(512); // Ініціалізувати EEPROM
DisplayWaIT_CARD(); // Запускаємо функцію, заставка дисплея.
EEPROMreadUIDcard(); // Запускаємо функцію, для перезаписи масива

digitalWrite(PiezoPin, HIGH), delay(100), digitalWrite(PiezoPin, LOW);
// Подача писк при запуске.
}

```

```

void loop() {
digitalWrite(SLN_PIN, HIGH); // Подати на пін реле напруги
if (digitalRead(BTN_PIN) == LOW) { // якщо на кнопці логічний 0
digitalWrite(SLN_PIN, LOW); //Подача на пін реле логічний 0
lcd.print(0, 0, " BUTTON UNLOCK "); // Вивести на дисплей
lcd.print(0, 1, " DOOR OPEN  "); // Вивести на дисплей
// Подати звук
digitalWrite(PiezoPin, HIGH), delay(200), digitalWrite(PiezoPin, LOW);
delay(2000);
DisplayWaIT_CARD(); // Запускаєм функцію, заставка дисплея.
}
// Пошук нової NFC карти
if (beginCard == 0) {
if ( ! mfrc522.PICC_IsNewCardPresent()) {
Blynk.run();
return;
}
// читаємо карту
if ( ! mfrc522.PICC_ReadCardSerial()) {
Blynk.run();
return;
}
}
}

// видача серійного номера картки 'uid'
for (byte i = 0; i < mfrc522.uid.size; i++) {
uidDecTemp = mfrc522.uid.uidByte[i];
uidDec = uidDec * 256 + uidDecTemp;
}
}

```

```

if (beginCard == 1 || LockSwitch > 0)EEPROMwriteUIDcard(); //uidDec ==
adminID

if (LockSwitch == 0) {
// готуємося для перебору масиву carduideepromread
for (ARRAYindexUIDcard = 0; ARRAYindexUIDcard <= 9;
ARRAYindexUIDcard++) {
// перебираємо весь масив carduideepromread, фільтруємо від 0
if (CardUIDeEPROMread[ARRAYindexUIDcard] > 0) {
// якщо виявлено збіг з піднесеної картою
if (CardUIDeEPROMread[ARRAYindexUIDcard] == uidDec) {
lcd.print(0, 0, "CARD ACCESS OPEN"); // Вивести на дисплей
lcd.print(3, 1, uidDec); // Вивести на дисплей uidDec картки
digitalWrite(SLN_PIN, LOW); //Відкрити замок
// Подати звуковий сигнал
digitalWrite(PiezoPin, HIGH), delay(200), digitalWrite(PiezoPin, LOW);
delay(2000);
break;
}
}
}

if (ARRAYindexUIDcard == 10) { // Значить, що ми перебрали весь масив від
//0 до 9, і не виявили збігів з піднесеної "Card UID".
// Тобто поднесення карта відсутня в базі, виводимо на дисплей "не
// знайдено карта" і № "Card UID" карти.
lcd.print(0, 0, " Card not Found "); // Вивести на дисплей
lcd.print(0, 1, " "); // Вивести на дисплей
lcd.print(0, 1, "ID : "); // Вивести на дисплей
lcd.print(5, 1, uidDec); // Вивести на дисплей
for (int i = 0; i <= 2; i++)delay(100), digitalWrite(PiezoPin, HIGH), delay(100),

```

```

digitalWrite(PiezoPin, LOW); // Подати звуковий сигнал
digitalWrite(SLN_PIN, HIGH); //Заблокувати двері
delay(2000);
}

ARRAYindexUIDcard = 0;
DisplayWaIT_CARD(); // Запускаємо функцію, заставка дисплея.
}
Blynk.run();
}
// Віджет кнопки на вивід V1 що ініціює запис нових карток.
BLYNK_WRITE(V1) {
int a = param.asInt();
if (a == 1) beginCard = 1; else beginCard = 0;
}
// Віджет кнопки на вивід V2 що надає змогу пропустити запис до комірки.
BLYNK_WRITE(V2) {
int a = param.asInt();
if (a == 1) {
skipCard = 1;
if (EEPROMstartAddr / 5 < 10) EEPROMwriteUIDcard();
} else {
skipCard = 0;
}
}
// Віджет кнопки на вивід V3 що відкриває двері.
BLYNK_WRITE(V3) {
int a = param.asInt();
if (a == 1) {
digitalWrite(SLN_PIN, LOW); //Відкрити

```



```

lcd.print(0, 0, " APP UNLOCK OK "); // Вивести на дисплей
lcd.print(0, 1, " DOOR OPEN  "); // Вивести на дисплей
digitalWrite(PiezoPin, HIGH), delay(200), digitalWrite(PiezoPin, LOW);
delay(2000); // Подати звуковий сигнал
DisplayWaIT_CARD();
}
}
// Робимо функцію, записи в EEPROM "Card UID" користувачів
// void EEPROMwriteUIDcard () {

if (LockSwitch == 0) { // Если находились в режиме управления.
lcd.print(0, 0, " START REC CARD "); // Вивести на дисплей
lcd.print(0, 1, "PLEASE TAG CARDS"); // Вивести на дисплей
delay(500); // зачекати 500 Мілі секунд
}

// Для пропуску записів в комірку пам'яті.
if (LockSwitch > 0) { // Якщо знаходимося в "режимі запису".
if (skipCard == 1) { //uidDec == adminID
lcd.print(0, 0, " SKIP RECORD  "); // Вивести на дисплей
lcd.print(0, 1, "          "); // Вивести на дисплей
lcd.print(0, 1, " label : "); // Вивести на дисплей
// Виводимо № пропущеної осередки пам'яті.
lcd.print(11, 1, EEPROMstartAddr / 5);
// Пропускаємо запис в комірку пам'яті, якщо не хочемо записувати туди
//"Card UID".
EEPROMstartAddr += 5;
skipCard = 0;
} else {
// "Card UID" / № карти це "довге число", яке не поміститься в одну комірку
//пам'яті EEPROM.

```

```

// розрубати "довге число" на 4 частини, і шматками, запишемо його в 4
// комірки EEPROM. Починаємо запис з адреси EEPROMstartAddr.
Serial.println("writeCard");
EEPROM.write(EEPROMstartAddr, uidDec & 0xFF);
EEPROM.write(EEPROMstartAddr + 1, (uidDec & 0xFF00) >> 8);
EEPROM.write(EEPROMstartAddr + 2, (uidDec & 0xFF0000) >> 16);
EEPROM.write(EEPROMstartAddr + 3, (uidDec & 0xFF000000) >> 24);
EEPROM.commit();
// Записали !
delay(10); // зачекати 10 Мілі секунд
lcd.print(0, 1, "          "); // Вивести на дисплей
lcd.print(0, 0, "RECORD OK! IN  "); // Вивести на дисплей
lcd.print(0, 1, "MEMORY : "); // Вивести на дисплей
lcd.print(9, 1, EEPROMstartAddr / 5); // Виводимо № записаного елементу
пам'яті.
EEPROMstartAddr += 5; // Додаємо 5 до стартового комірки запису.
delay(500); // зачекати 500 Мілі секунд
}
}

// Розблоковуємо режим запису, і заблокуємо режим управління.
LockSwitch++;
if (EEPROMstartAddr / 5 == 10) { // якщо дійшли до 10.
lcd.clear();
lcd.print(0, 0, "RECORD FINISH"); // Вивести на дисплей
delay(2000); // зачекати 2000 Мілі секунд
EEPROMstartAddr = 0;
uidDec = 0;
ARRAYindexUIDcard = 0;
EEPROMreadUIDcard(); // Запускаємо функцію, для перезапису масиву

```

```

CardUIDeEPROMread, даними з EEPROM.
}
}
// Робимо функцію, перезапису масиву CardUIDeEPROMread, даними з
//EEPROM.
void EEPROMreadUIDcard() {
for (int i = 0; i <= 9; i++) {
byte val = EEPROM.read(EEPROMstartAddr + 3);
CardUIDeEPROMread[ARRAYindexUIDcard] =
(CardUIDeEPROMread[ARRAYindexUIDcard] << 8) | val;
val = EEPROM.read(EEPROMstartAddr + 2);
CardUIDeEPROMread[ARRAYindexUIDcard] =
(CardUIDeEPROMread[ARRAYindexUIDcard] << 8) | val;
// збільшуємо EEPROMstartAddr на 1.
val = EEPROM.read(EEPROMstartAddr + 1);
CardUIDeEPROMread[ARRAYindexUIDcard] =
(CardUIDeEPROMread[ARRAYindexUIDcard] << 8) | val;
val = EEPROM.read(EEPROMstartAddr);
CardUIDeEPROMread[ARRAYindexUIDcard] =
(CardUIDeEPROMread[ARRAYindexUIDcard] << 8) | val;

ARRAYindexUIDcard++; // збільшуємо на 1.
EEPROMstartAddr += 5; // збільшуємо на 5.
}
}

ARRAYindexUIDcard = 0;
EEPROMstartAddr = 0;
uidDec = 0;
LockSwitch = 0;

```

```
DisplayWaiT_CARD();  
}  
// Робимо функцію, заставка дисплея.  
void DisplayWaiT_CARD() {  
  lcd.print(0, 0, " ATTACH THE "); // Вивести на дисплей  
  lcd.print(0, 1, " CARD "); // Вивести на дисплей  
}
```