Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
(повне найменування факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
бакалавра
(ступінь вищої освіти)

на тему Розробка багатофункціонального пристрою контролю доступу до приміщення
(назва теми)

Виконав: студент 4 курсу, групи КНТ-519

Спеціальності
123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерна інженерія

КАЛІСТРАТОВ О.Г.
(ПРІЗВИЩЕ та ініціали)

Керівник ІЛЬЯШЕНКО М.Б.
(ПРІЗВИЩЕ та ініціали)

Рецензент СТЕПАНЕНКО О.О.
(ПРІЗВИЩЕ та ініціали)

2023

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
Кафедра Комп'ютерні системи та мережі
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерна інженерія
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ
Завідувач кафедри Кудерметов Р.К.
_____
«_____»_____2023року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Калістратов Олексій Геннадійович
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка багатофункціонального пристрою контролю доступу до приміщення
керівник проєкту (роботи) к. т. н., доцент, ІЛЬЯШЕНКО Матвій Борисович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від «24» березня 2023 року № 71
2. Строк подання студентом проєкту (роботи) 25 травня 2023 року
3. Вихідні дані до проєкту (роботи)програмні компоненти: Arduino IDE для написання коду

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз технічного завдання
Проєктний розділ
Виготовлення системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Пл1 – Елементи системи;
Пл2 – Підключення модуля RC533 до Arduino Uno;
Пл3 – Основні етапи роботи пристрою;
Пл4 – Повна схема контролю доступу;

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
	ІЛЬЯШЕНКО М.Б., к. т. н., доцент		
Нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання «8» травня 2023року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
	Аналіз технічного завдання	1 тиждень	виконано
	Аналіз структури роботи	1 тиждень	виконано
	Вибір відповідної технології для створення системи	2 тиждень	виконано
	Підбір комплектуючих елементів для створення системи, схеми підключення	2 тиждень	виконано
	Написання програмного коду	3 тиждень	виконано
	Реалізація проекту	3 тиждень	виконано
	Оформлення проекту	4 тиждень	виконано
	Проходження нормоконтролю	4 тиждень	виконано
	Проектування програмних засобів	4 тиждень	виконано
	Захист роботи	4 тиждень	виконано

Студент(ка) _____ **Олексій КАЛІСТРАТОВ**
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ **Матвій ІЛЬЯШЕНКО**
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

ПЗ: 60 с., 19 рис., 3 табл., 19 джерел.

ІОТ, ПРИСТРІЙ, СИСТЕМА , СИСТЕМА ДОСТУПУ

Тема даної дипломної роботи – «Розробка багатофункціонального пристрою контролю доступу до приміщення», у межах якої було розглянуто основні засади розробки та створення системи, що дозволяє контролювати доступом до закритим приміщенням.

Об'єктом дослідження є макет пристрою контролю доступу до приміщення на мікроконтролерному управлінні.

Мета роботи полягає у виготовленні макету пристрою контролю доступу на мікроконтролерному управлінні.

Для того, щоб вирішити поставлену мету роботи, були сформульовані основні завдання:

- огляд існуючих рішень;
- розробка схеми пристрою;
- вибір необхідних компонентів;
- завдання алгоритму роботи;
- створення програми роботи пристрою;
- виготовлення макету пристрою;
- налагодження програми та макета.

Для створення наочних схем підключення різних елементів використовувалися програмні рішення Autodesk Circuits та Fritzing.

Результатом дипломної роботи є макет пристрою контролю доступу в приміщення на мікроконтролерному управлінні.

ЗМІСТ

Перелік графічного матеріалу:

Пл1 – Елементи системи;

Пл2 – Підключення модуля RC533 до Arduino Uno;

Пл3 – Основні етапи роботи пристрою;

Пл4 – Повна схема контролю доступу;

ВСТУП

Електронний замок — спеціальний електронний пристрій, необхідний для того, щоб запобігти доступу до закритого приміщення сторонніх осіб, або обмежити вихід із приміщення. Система контролю приймає рішення про дозвіл на доступ до приміщення на основі сигналів від різних пристроїв: зчитувачів магнітних карт, штрих-кодів, датчиків контактної пам'яті, біометричних датчиків, набірної клавіатури, зчитувачів магнітних карт, дистанційного керування та інших різноманітних датчиків. У більшості випадків електронний замок є складовою системою контролю доступу в приміщення. Як механізми, що перешкоджають у доступі в приміщення, використовуються електромеханічні та електромагнітні запірні пристрої. Використання мікроконтролера дозволить спростити основні маніпуляції із системою контролю доступу та самим електронним замком.

В рамках даної роботи передбачається розробити, створити і налагодити невелику діючу модель контролю доступу на мікроконтролерному управлінні. Така модель дозволить практично продемонструвати роботу електронного замку та супутніх йому елементів.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Формулювання актуальності, мети та задач

В даний час електронні системи контролю доступу в приміщення набувають все більшого поширення в житті величезної кількості людей. Електронні замки використовуються там, де заборонено знаходитись стороннім, наприклад, у складських приміщеннях підприємств, у підсобних приміщеннях у магазинах та великих супермаркетах. Незважаючи на це, електронні системи контролю доступу є досить дорогим обладнанням для звичайної людини. Але якщо існують рішення, які б дозволили широкому колу населення використовувати електронні замки у різних цілях.

Виходячи з вищесказаного, розробка функціонального пристрою контролю доступу в приміщення на мікроконтролерному управлінні є актуальним завданням.

Актуальність розроблення багатофункціонального пристрою контролю доступу в приміщення

Безпека у сучасному світі забезпечення безпеки стає дедалі важливішим завданням. Пристрої контролю доступу відіграють ключову роль у забезпеченні безпеки приміщень, обмеженні несанкціонованого доступу та запобіганні злочинним діям.

Зручність та ефективність багатофункціональний пристрій контролю доступу дає змогу керувати доступом користувачів до приміщень, що забезпечує зручність та ефективність роботи. Він здатен автоматично реєструвати та відстежувати доступ, спрощуючи процес контролю та обліку.

Гнучкість і масштабованість розробка багатофункціонального пристрою дає змогу створювати системи, які можуть бути адаптовані під різні типи приміщень і вимоги. Такий пристрій може інтегруватися з іншими системами безпеки та розумним будинком, забезпечуючи комплексний підхід до контролю доступу.

Технологічний прогрес. З постійним розвитком технологій, включно з безконтактними картками, біометричною ідентифікацією та мережевими зв'язками, розробка багатофункціонального пристрою контролю доступу стає актуальною. Вона дає змогу використовувати новітні технології для забезпечення високого рівня безпеки та зручності.

Зниження ризиків багатофункціональний пристрій контролю доступу допомагає знизити ризики несанкціонованого доступу, крадіжки, розбою та інших злочинів. Це особливо важливо для організацій, де зберігаються цінні матеріали, документи та дані.

Керування ресурсами багатофункціональний пристрій дає змогу ефективно керувати доступом до приміщень, контролювати використання ресурсів, як-от енергія та вода. Це сприяє економії та зниженню витрат на експлуатацію приміщень.

Загалом розробка багатофункціонального пристрою контролю доступу до приміщення актуальна в сучасному суспільстві, де безпека, зручність і ефективність стають дедалі важливішими факторами. Це допоможе запобігти злочинним діям, забезпечити безпеку персоналу та активів, а також оптимізувати управління ресурсами та процесами в приміщенні.

Метою даної роботи є: виготовлення макету пристрою контролю доступу на мікроконтролерному управлінні.

Для досягнення цієї мети поставлені та виконані такі завдання:

- огляд існуючих рішень;
- розробка схеми пристрою;
- вибір необхідних компонентів;
- завдання алгоритму роботи;
- створення програми роботи пристрою;
- виготовлення макету пристрою;
- налагодження програми та макета.

1.2 Аналіз вихідних даних та відомих рішень

На сьогоднішній день є кілька варіацій замків, яким не потрібен ключ у звичному для нас розумінні для того, щоб відкрити або закрити двері. Є кілька типів замків - з магнітним ключем, представленим у вигляді брелока або картки, схожої на банківську, або з кодовою комбінацією, які встановлюються прямо на двері, про які піде нижче.

Сучасні кодові замки можна класифікувати за декількома ознаками: за типом установки, способом керування механізмом замикання дверей, можливості змінювати кодову комбінацію.

За типом кодові замки поділяються на навісні та врізні замки. Навісні замки використовуються в основному для сараїв, гаражів, складських приміщень, і в інших подібних випадках, але для житлових будинків і квартир краще використовувати врізні кодові замки (рисунок 1.1), які надійніші за навісні через те, що робочі механізми такого замку розташовуються всередині дверного полотна, захищаючи його від можливості розбору та злому ззовні. Навісні замки можна використовувати як додатковий захист.

За принципом управління замикаючим механізмом замки можна розділити на механічні, приклад показаний на рисунку 1.2, та електронні. В даний час механічні замки починають втрачати свою привабливість, хоча не можна з упевненістю говорити, що їх вже починають сильно витісняти електронні моделі, адже час механіки в нашій країні ще довго триватиме, і важко точно сказати, коли електроніка почне домінувати сфері замків.

Рисунок 1.1 – Замок врізного типу

Незважаючи на це, механічні замки мають свої недоліки, наприклад, вони швидше виходять з ладу, внаслідок постійного їх використання також серйозною проблемою є те, що до цих замків досить легко підібрати потрібну комбінацію для відкриття, в порівнянні з іншими видами.

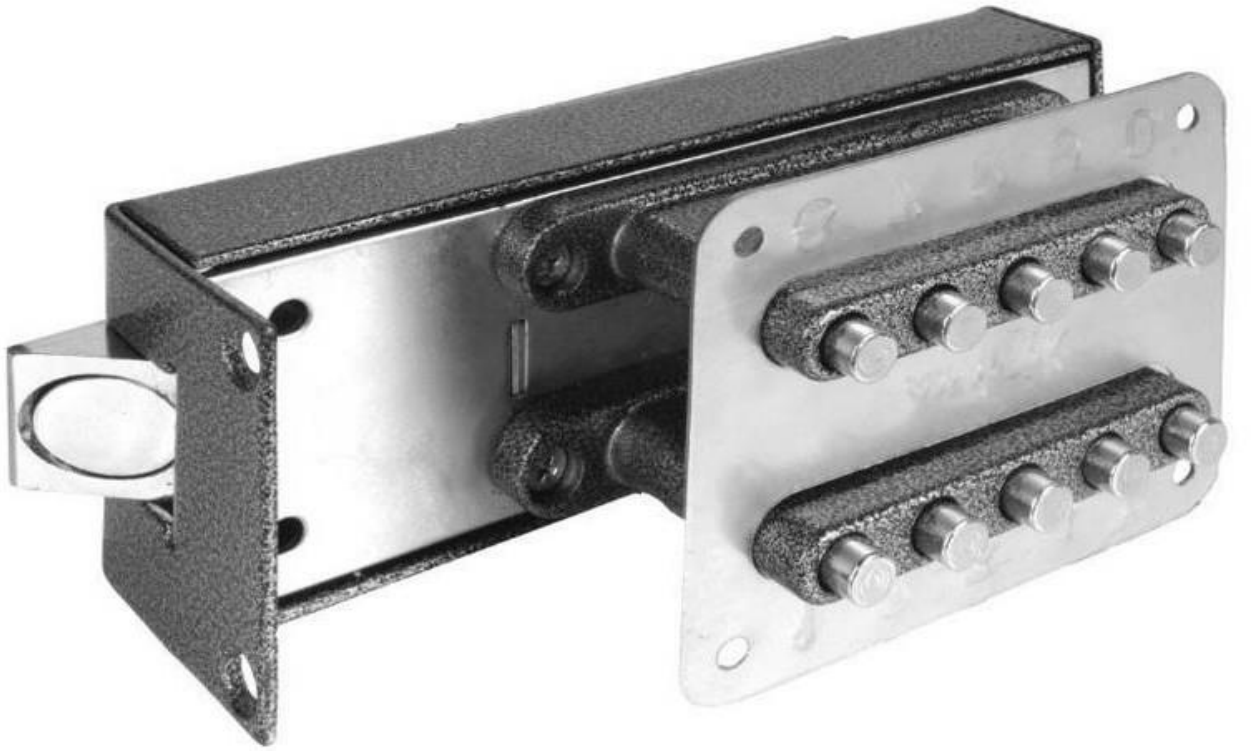


Рисунок 1.2 – Механічний кодовий замок

Механічні замки мають кодовий кнопковий механізм або поворотний. На практиці кнопкові замки показують себе найгірше. Постійні натискання на кнопки стирають їхній верхній шар, і вони починають западати, що дає можливість будь-якій людині легко підібрати правильну комбінацію. Поворотний механізм, порівняно з кнопковим, має незаперечну перевагу – визначити, як і скільки разів і в який бік потрібно повернути ручку не вийде, незважаючи на те, наскільки затерті написи або як часто цим замком користувалися. Існують також електронні кодові замки, які заслуговують на окрему увагу.

Електронні кодові замки мають одну істотну перевагу перед механічними – блок управління може знаходитися на відстані від основного пристрою. Його можна встановити будь-де – це основа принципу замку – невидимки. Коли немає можливості бачити те, з чим доведеться мати справу, стає проблематично розкрити такий пристрій. Крім того, сучасний кодовий замок – це високотехнологічний пристрій, який керується за допомогою мікропроцесора, а це мільйони різних комбінацій. Усі електронні кодові замки можна розділити на

наступні категорії.

Кнопкові кодові замки із електронним керуванням. Вони набули найбільшого поширення, але в той же час це вразлива група замикаючих пристроїв. Причина, як і у випадку механічних замків, та ж - кнопки, що стираються і западають. Здебільшого їх встановлюють там, де не потрібно зберігати особливо цінні предмети. Їх використовують для запобігання проникненню в приміщення, наприклад, у під'їзди, складські та підсобні приміщення в невеликих підприємствах або магазинах. Найчастіше пульт управління і сам замок виготовляються одним блоком, але є моделі з роздільним оснащенням. З розвитком технологій кнопкові кодові замки отримали своє логічне продовження у вигляді замків із сенсорною панеллю. Переваги перед кнопками тут очевидні, але є серйозна проблема з тим, що на марких сенсорних екранах при натисканні залишаються відбитки пальців, які стають помітними під певним кутом огляду. Для уникнення злому комбінації за відбитками потрібні різні хитрощі, на які йдуть виробники замків із сенсорною панеллю. Одним із способівприховування комбінації є введення додаткових непотрібних символів спочатку або наприкінці основного коду.

Кодові замки із магнітним носієм комбінації. Це дуже надійні замки, розкрити які можна, якщо отримати сам магнітний носій. У цій ролі можуть виступати: пластикова картка, невеликий брелок, пульт дистанційного керування, що передає код приймачеві радіосигналом або сигналом в інфрачервоному спектрі, що легко перехоплюється і декодується.

Комбіновані замки (рис. 1.3). Даний вид замків найпоширеніший на сьогоднішній день. Потрапити всередину приміщення можна тільки при використанні послідовно декількох різних пристроїв, наприклад, кодової комбінації і пластикової карти, що робить комбіновані замки дуже надійною системою. Їх особливістю є те, що розблокувати двері неможливо лише за допомогою одного ключа, потрібно пройти всі послідовні дії, необхідні для відкриття, а також те, що ключі від інших виробників будуть ігноруватися при спробі їх використання.

Рисунок 1.3 – Електронний кодовий замок із сенсорною панеллю введення виробництва компанії Samsung

Пристрій електронного замку складається з чотирьох частин:

– замикаючий механізм. Щоб він відкрився чи закрився, на нього подають короткий електричний імпульс. Коли кодова комбінація співпадає із заданою, замок відкриється.

– зчитувач коду чи пульт управління. Це зчитуючий пристрій – електроніки

управління у ньому немає. За допомогою цього пристрою вводиться код і спрямовується в блок керування.

– блок керування. Він розпоряджається питанням, подавати чи не подавати імпульс для спрацьовування замикаючого механізму. В основному замки закриваються автоматично як, в принципі, і будь-який інший механічний замок, що закривається.

– джерело безперебійного живлення (ДБЖ). Наявність ДБЖ у таких замках обумовлена тим, що під час відключення електроенергії потрапити до будинку буде важко, адже за відсутності електрики замки автоматично блокуються. З його допомогою кодовий електронний замок опрацює автономно кілька діб. Зазвичай ДБЖ знаходиться в таємному місці, там же, де і блок управління.

У мережі Інтернет також можна знайти різні схемотехнічні рішення реалізації електронних кодових замків, наприклад, як у рисунку 1.4. Дана схема являє собою пристрій, що складається з: двох КМОП мікросхем 561ЛА7 та однієї 561ЛЕ5, транзистори VT1-VT3 - КТ361, або КТ3107, транзистор VT4 -КТ315, транзистор VT5 - КТ815. Вторинна обмотка трансформатора Т1 розрахована на 12 вольт. Трансформатор Т1 вибирається достатньої потужності, що забезпечує спрацьовування виконавчого пристрою, діоди VD3-VD7 вибираються випрямляючими FR107, що забезпечують достатній струм навантаження виконавчого пристрою. Діоди VD8-VD20 – малопотужні імпульсні КД521. Як акумуляторна батарея використовується малогабаритна лужна батарея, що використовується виточниках безперебійного харчування.

Набір коду здійснюється за допомогою панелі кнопки SA1.

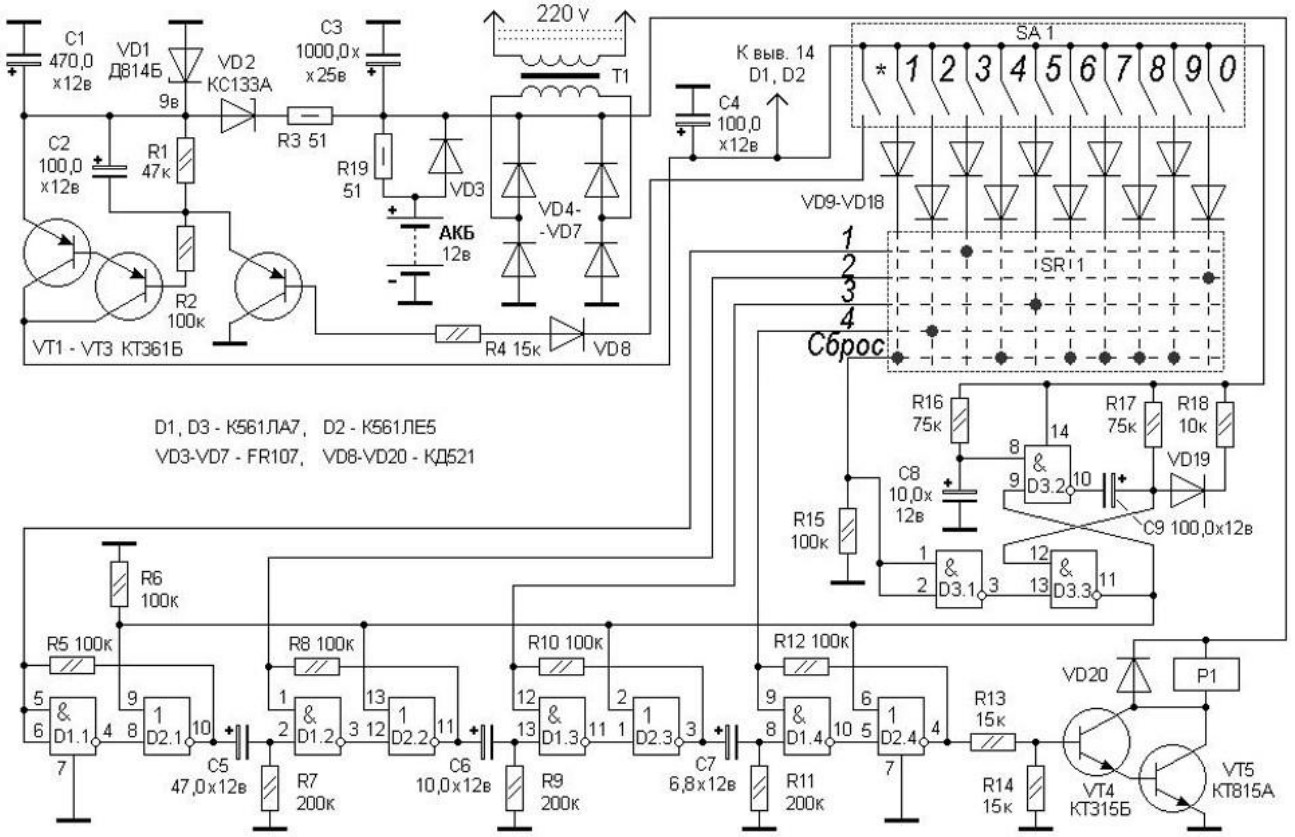


Рисунок 1.4 – Схема електрична важлива електронного кодового замку

Заявленими перевагами даної схеми в порівнянні з іншими схемотехнічними рішеннями, доступними в мережі Інтернет є: висока захищеність від зламування обманними кнопками, при натисканні на які замок блокується на деякий час, кнопкова панель розташовується окремо від основного пристрою, завдяки чому зламування за допомогою вимірювальної апаратури стає неможливе, живлення від акумулятора на 12 вольт, або від мережі змінного струму 220 вольт.

Незважаючи на це, представлена вище схема кодового замку має свої недоліки як електронний замок: чутливість до електромагнітних перешкод, важких погодних умов, вимога до безперебійної подачі живлення. Але головним фактором є те, що для створення подібних схемотехнічних рішень потрібні глибокі пізнання в схемотехніці і в мікросхемної логіки, потрібен точний підбір елементів для правильної роботи, тому через перерахований вище досить важко розглядати цей тип замку як навчального ознайомлення. Тому основою для цієї роботи було обрано електронний замок на мікроконтролерному управлінні.

Для того щоб реалізувати пристрій контролю доступу на мікроконтролерному управлінні, в якості вихідних даних були обрані способи управління за допомогою кодової комбінації, що реалізується за допомогою кнопок, а також ідентифікація радіочастотна, реалізована за допомогою спеціально розроблених для роботи з мікроконтролерами модулями радіочастотної ідентифікації. Розглянемо докладніше основні засади організації радіочастотної ідентифікації.

RFID – радіочастотна ідентифікація є фундаментальною та недорогою технологією, що здійснює бездротову передачу даних. Раніше ця технологія не так часто використовувалася в індустрії у зв'язку з відсутністю стандартизації у виробничих компаніях. RFID – технології ефективніші та надійніші, порівняно з іншими. З RFID - технологією бездротова автоматична ідентифікація набуває дуже специфічного вигляду: об'єкт, місцезнаходження, або індивід маркуються унікальним ідентифікаційним кодом, що міститься в RFID - мітці, яка якимось чином прикріплена або втиснута в об'єкт. Радіочастотна ідентифікація є не окремим продуктом, а повноцінною системою. Типова RFID - система включає три базові елементи: RFID-мітку (транспондер), зчитувач (трансівер, запитувач) і серверну програму (базу даних), яка вимагає підтримки комп'ютерної мережі. Програмне забезпечення використовується для управління, контролю, оперування, обробки, ведення обліку різних користувачів. Цифрова система блокування дверей здійснюється та управляється за допомогою RFID – зчитувача, який проводить перевірку та автентифікацію користувача та автоматично відчиняє двері. Він також зберігає дані про реєстрацію користувача. Дуже важливо провести аутентифікацію користувача до відкриття приміщення, що охороняється, і радіочастотна ідентифікація дозволяє зробити це. Система дозволяє користувачеві реєструватися у швидких, безпечних зручних умовах. Система блокування дверей відкриває їх тільки тоді, коли користувач помістить свою мітку на зчитувач, а дані користувача зрівняються з тими, що зберігаються в базі даних. Радіо частотна ідентифікація контролює відкриття та закриття дверей. Залежно від джерела електричної енергії, теги RFID класифікуються як активні або пасивні. Активні

мітки використовують батарею для живлення та передачі інформації мітки на запит зчитувач. Однак ці теги дуже дорогі та рідко використовуються. З іншого боку, пасивні мітки отримують енергію від зчитувача живлення їх схеми. Ці мітки дуже рентабельні і, отже, більшість програм використовують їх. Головними перевагами пасивної технології є низька вартість та невеликі габарити. Пасивна мітка RFID передає інформацію зчитувачу, коли вона потрапляє в електромагнітне поле, що генерується зчитувачем. Явище ґрунтується на законі електромагнітної індукції Фарадея. Струм, що протікає через котушку зчитувача, створює магнітне поле, яке з'єднується з котушкою транспондера, створюючи тим самим струм в котушці транспондера. Потім котушка транспондера змінює цей струм, змінюючи навантаження на свою антену. Ця зміна фактично є модульованим сигналом, який приймається котушкою зчитувача за допомогою взаємної індукції між котушками. Котушка зчитувача декодує цей сигнал та передає його на комп'ютер для подальшої обробки. Додаткове підключення антени дозволяє зменшити розмір пристрою.

Існує кілька типів класифікації міток. Широко відома та поширена поділяє мітки на чіпові (рисунок 1.5) та безчіпові (безмікросхемні), зображені на рисунку 1.6.

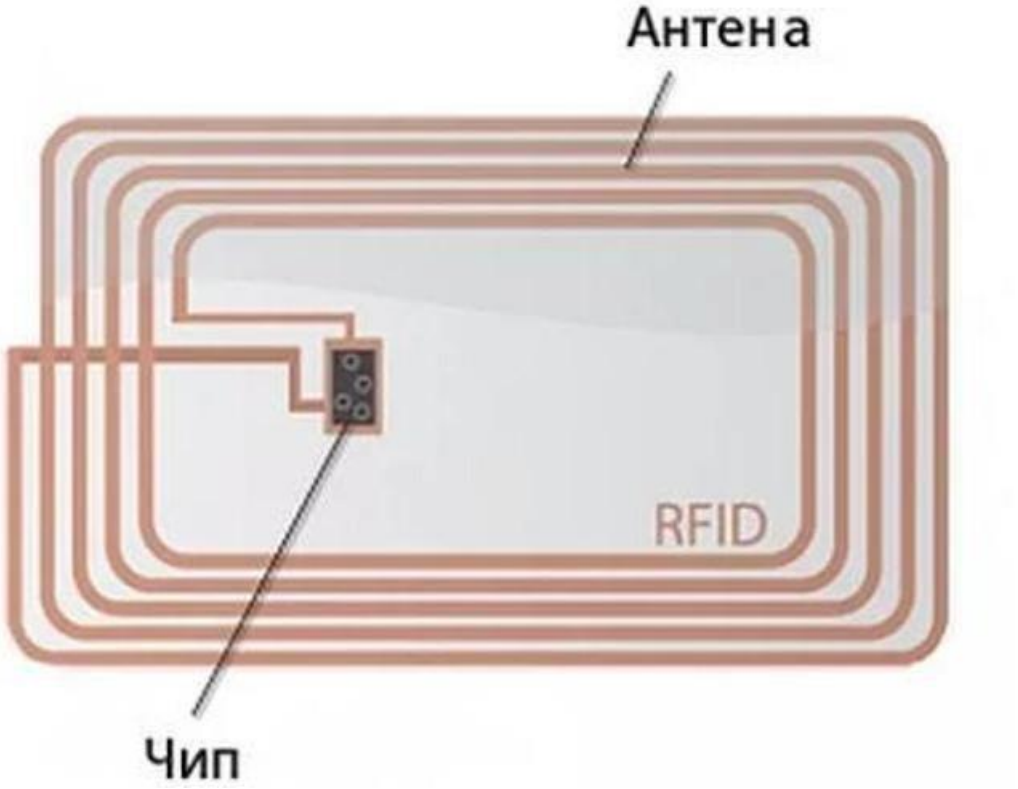


Рисунок 1.5 – RFID-мітка

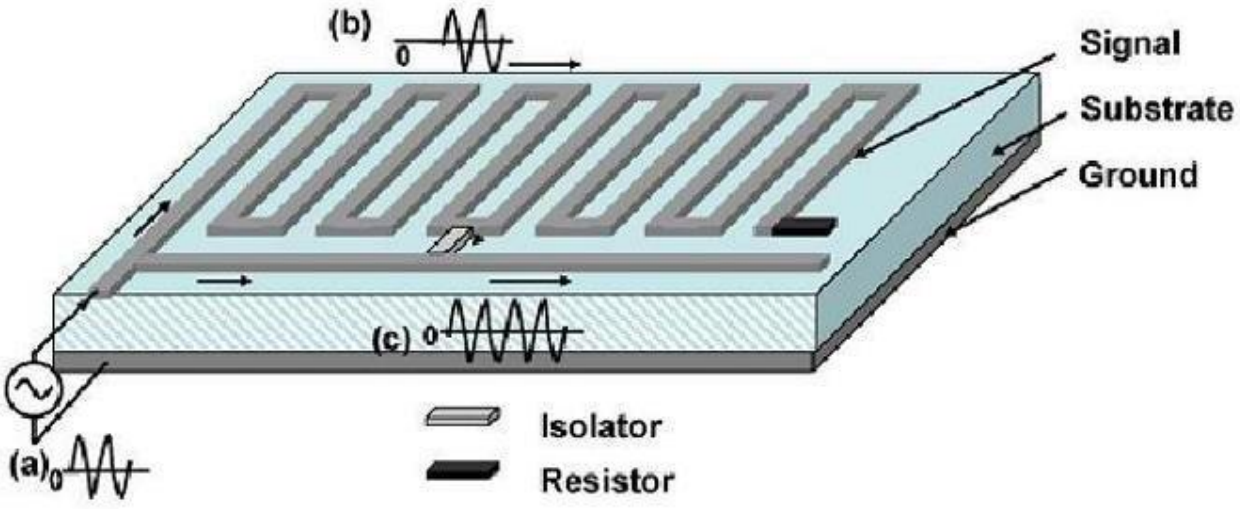


Рисунок 1.6 - Безчіпова RFID-мітка

Чипові мітки містять інтегральну мікросхему – чіп, а безчіпові – ні. Друга класифікація поділяє типи міток на пасивні, напівактивні та активні. До складу пасивних міток не входить вбудований джерело живлення та активний передавач; напівактивні мітки містять елемент живлення, але не мають активного передавача;

активні мітки містять обидва елементи. У ще одній класифікації мітки поділяються на лише зчитувані (read only) і зчитувані/записуючі (read/write). Тільки мітки, що зчитуються, отримують свій ідентифікаційний код на виробництві. Пам'ять даного типу міток або читається (ROM) або одноразово програмована і багаторазово читається (WORM), тобто. під час роботи з WORM пам'яттю, ідентифікаційний код даного типу міток можна перезаписати один раз.

Теги, що зчитуються/записують, можуть багаторазово перепрограмуватися в процесі експлуатації, і на них можна записати додаткову інформацію, а не тільки серійний номер або код.

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

1.3 Бездротові технології Інтернету речей

Розвиток сучасних технологій призвів до виникнення різних інноваційних рішень у сфері безпеки та управління доступом. Одним з таких рішень є багатофункціональний пристрій контролю доступу, який заснований на застосуванні бездротових технологій. У цій статті ми розглянемо значущість і переваги використання бездротових технологій для розробки такого пристрою.

Роль бездротових технологій у розробці багатофункціонального пристрою контролю доступу

Бездротові технології відіграють ключову роль у розробці багатофункціонального пристрою контролю доступу в приміщення. Вони забезпечують гнучкість, масштабованість і зручність використання такого пристрою. Розглянемо основні аспекти використання бездротових технологій у цьому контексті.

Безконтактні технології. Безконтактні технології, такі як RFID (Radio

Frequency Identification) і NFC (Near Field Communication), дають змогу реалізувати систему контролю доступу без необхідності фізичного контакту з пристроєм. Це забезпечує зручність і швидкість проходу для користувачів, а також мінімізує знос і поломки, пов'язані з використанням механічних ключів.

Використання бездротових мереж, як-от Wi-Fi і Bluetooth, дає змогу встановлювати зв'язок між багатофункціональним пристроєм контролю доступу та іншими пристроями, як-от смартфони, планшети або комп'ютери. Це розширює функціональність і можливості пристрою, даючи змогу керувати ним віддалено, передавати дані та налаштовувати параметри роботи.

З розвитком мобільних технологій і застосунків, можна створити спеціальний мобільний застосунок, який дасть змогу користувачам керувати доступом до приміщень через багатофункціональний пристрій контролю доступу. Мобільні додатки забезпечують гнучкість, зручність використання і додаткові функціональні можливості, як-от надсилання повідомлень, звітів і логів.

Інтеграція багатофункціонального пристрою контролю доступу з хмарними технологіями дає змогу зберігати дані про доступ, налаштування та події в хмарі. Це забезпечує зручність резервного копіювання, віддаленого доступу до даних і централізоване управління кількома пристроями контролю доступу.

Переваги бездротових технологій у розробці багатофункціонального пристрою контролю доступу.

Використання бездротових технологій у розробці багатофункціонального пристрою контролю доступу має низку переваг.

Гнучкість і масштабованість: Бездротові технології дають змогу створювати гнучкі та масштабовані системи контролю доступу. Пристрої можуть бути легко додані або видалені з мережі, а система може бути адаптована під різні типи приміщень і вимоги.

Зручність використання: Безконтактні технології та мобільні додатки забезпечують зручність використання для користувачів. Вони можуть швидко і легко отримати доступ до приміщень без необхідності використовувати ключі або пам'ятні коди.

Безпека: Бездротові технології дають змогу реалізувати більш безпечні системи контролю доступу. Наприклад, використання біометричних даних, таких як відбитки пальців або розпізнавання обличчя, забезпечує високий ступінь ідентифікації та запобігає несанкціонованому доступу.

Інтеграція та розширюваність: Бездротові технології дають змогу інтегрувати багатофункціональний пристрій контролю доступу з іншими системами безпеки та розумним будинком. Це дає змогу створити комплексну та інтелектуальну систему управління доступом і безпекою.

Бездротові технології відіграють важливу роль у розробці багатофункціонального пристрою контролю доступу в приміщення. Вони забезпечують гнучкість, зручність використання, безпеку та можливості інтеграції з іншими системами. Завдяки цим технологіям, багатофункціональні пристрої контролю доступу стають ефективним інструментом для забезпечення безпеки та управління доступом у різних типах приміщень.

2 ПРОЄКТНИЙ РОЗДІЛ

2.1 Розробка схеми і вибір необхідних компонентів

Arduino - це відкрита платформа для розробки електронних пристроїв і прототипування. Вона надає широкі можливості для створення різних проектів, починаючи від простих електронних пристроїв до складних автоматизованих систем. У цій статті ми розглянемо процес розроблення схеми та вибору необхідних компонентів для Arduino проєкту.

Перед початком розробки схеми Arduino необхідно визначити вимоги проекту. Які функції має виконувати пристрій? Які сенсори, актуатори або інші компоненти будуть використовуватися? Необхідно також врахувати фізичні обмеження, як-от розміри та живлення.

Для розробки схеми Arduino необхідно ознайомитися з документацією та ресурсами, доступними для обраної платформи Arduino. Вивчіть специфікації плати, схеми під'єднання та документацію щодо бібліотек і компонентів, які ви плануєте використовувати. Це допоможе вам краще зрозуміти можливості та обмеження вашого проєкту.

На цьому етапі ви можете почати розробляти електричну схему для вашого проєкту. Визначте, які компоненти будуть використовуватися і як вони будуть підключені до плати Arduino. Розташуйте компоненти на схемі та врахуйте правила під'єднання, як-от правильна напруга та положення контактів.

Після розробки схеми необхідно вибрати необхідні компоненти для вашого проекту. Це може включати в себе різні сенсори, актуатори, світлодіоди, реле та інші компоненти залежно від вимог вашого проєкту. Зверніть увагу на параметри компонентів, такі як напруга, струм, інтерфейси та сумісність з Arduino.

Перед під'єднанням компонентів до плати Arduino необхідно перевірити їхню сумісність і підтримку. Переконайтеся, що вибрані компоненти мають драйвери або бібліотеки, які можуть бути використані з Arduino. Підключіть компоненти до відповідних контактів Arduino, дотримуючись схеми підключення.

Після успішного підключення компонентів до Arduino необхідно розробити програму для керування ними. Використовуйте Arduino IDE або інше середовище розробки для написання коду, який керуватиме компонентами згідно з вимогами вашого проєкту. Завантажте програму на плату Arduino і проведіть тестування, щоб переконатися, що всі компоненти працюють правильно.

У процесі тестування ви можете зіткнутися з проблемами або недоліками в проєкті. Проведіть налагодження і поліпшіть свою схему і програму, щоб виправити проблеми. Ітеративно повторюйте процес тестування і поліпшення, поки ваш проєкт не працюватиме оптимально.

Розробка схеми і вибір необхідних компонентів для Arduino проекту - це важливий етап у створенні електронних пристроїв. Правильне планування, вивчення документації, вибір відповідних компонентів і тестування допоможуть створити функціональний і ефективний пристрій. Дотримуйтесь зазначених

кроків і не бійтеся експериментувати, щоб досягти бажаних результатів у вашому проєкті.

Як основу, яка контролюватиме всю роботу пристрою, був обраний мікроконтролер серії Arduino UNO R3, лицьова частина якого представлена на рисунку 2.1.

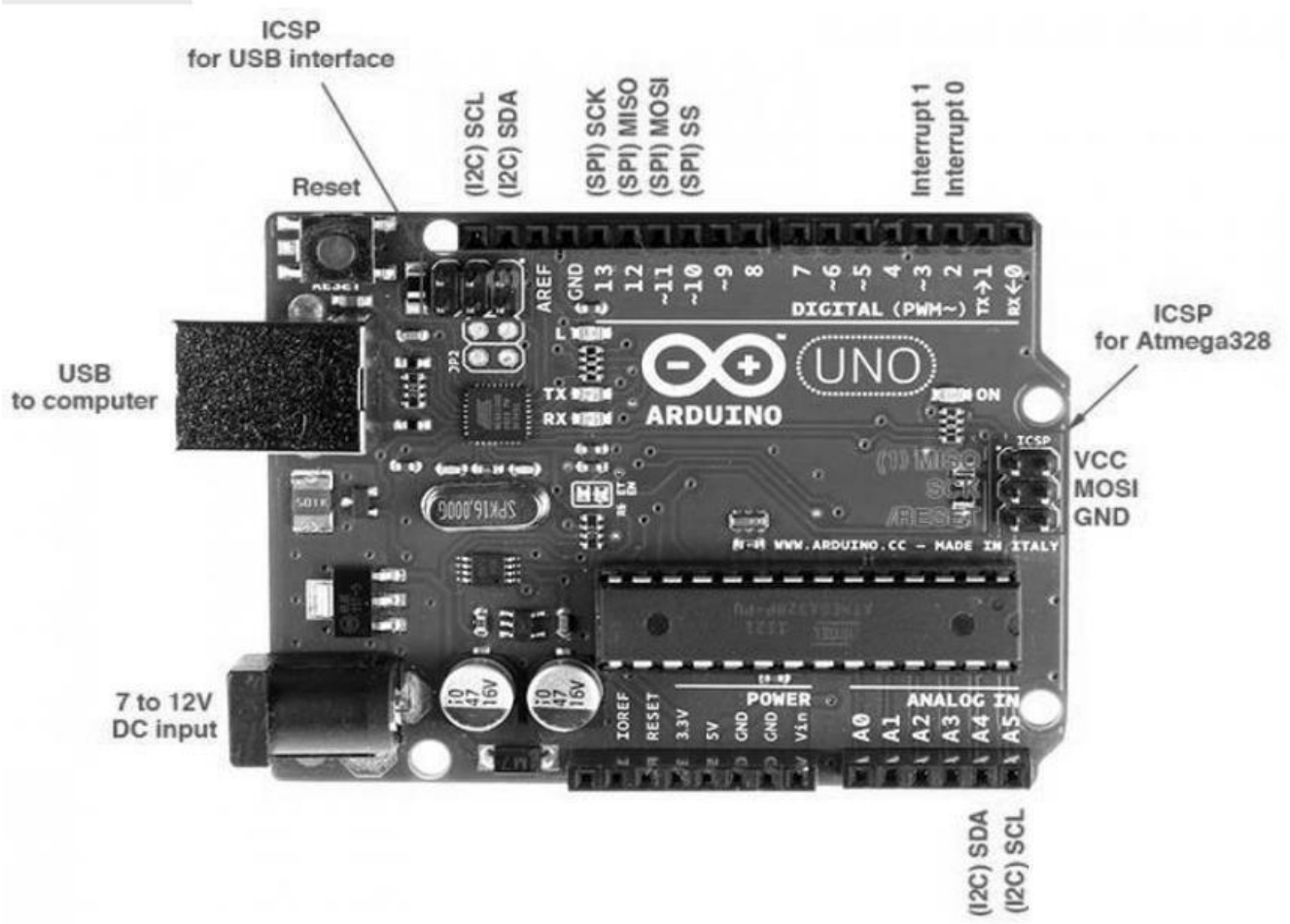


Рисунок 2.1 – Лицьова сторона Arduino Uno R3

Arduino Uno R3 – це пристрій на основі мікроконтролера ATmega328. До його складу входить: 14 цифрових входів/виходів під номерами 0-13 (з них 6 можуть використовуватися як ШІМ-виходи, на платі позначені зі знаком "~"), 6 аналогових входів A0-A5, кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрішньосхемного програмування (ICSP) та кнопка скидання. Для початку роботи потрібно подати живлення від AC/DC-адаптера або батарейки, або через USB кабель від персонального комп'ютера. Порт AREF

визначає опорну напругу аналогових входів. Порт IOREF дозволяє платам розширення підлаштовуватися під робочу напругу Arduino. Він необхідний для сумісності плат розширення як із 5 вольт (В) Arduino на базі мікроконтролерів AVR, так і з 3.3В платами Arduino Due.

Основні характеристики Arduino Uno R3 представлені нижче. Робоча напруга живлення 5В, рекомендована напруга живлення від 7 до 12В, гранична напруга живлення в діапазоні 6-20В. 14 цифрових входу/виходу, 6 аналогових входів, максимальний струм одного виводу дорівнює 40 мА, максимальний струм виведення 3.3V дорівнює 50 мА. У мікроконтролера є Flash-пам'ять на 32 кілобайт (КБ) (ATmega328), що використовується при створенні програм, з яких 0.5 КБ використовуються завантажувачем, а також електрично стирається енергонезалежна пам'ять EEPROM, що перезаписується, на 1 КБ. Тактова частота кварцового резонатора 16 МГц.

Для реалізації радіочастотної ідентифікації було обрано RFID-модуль RC522. Його параметри представлені нижче.

Напруга живлення 3,3 В, струм споживання не більше 30 мА, робоча смуга частот 13,55-13,57 МГц, зчитується на відстані 0-25 мм, фізичний розмір зчитувача 40 х 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 106, 212, 424, 848 кбіт/с. Шифрування Security Features Mifare classic™ (термін Mifare може лише компанія NXP Semiconductors, а також компанії, що мають ліцензію від NXP на виробництво чипів). Мітки MiFare Classic працюють на високочастотних радіохвилях, зокрема на частоті 13,56 МГц. Це та сама 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 з тих пір було продано понад мільярд міток у всьому світі. Діючи в якості систем контролю доступу та електронних гаманців, мітки привернули увагу дослідницьких груп, які провели численні дослідження щодо безпеки, які

пропонують мітки. MiFare Classic реалізують свій криптографічний алгоритм під назвою CRYPTO-1. Це потоковий шифр з 48-бітним секретним ключем, який використовується для забезпечення конфіденційності даних та взаємної автентифікації між міткою та зчитувачем.

Зовнішня схема пристрою для зчитування RFID-міток представлена рисунку 2.2.

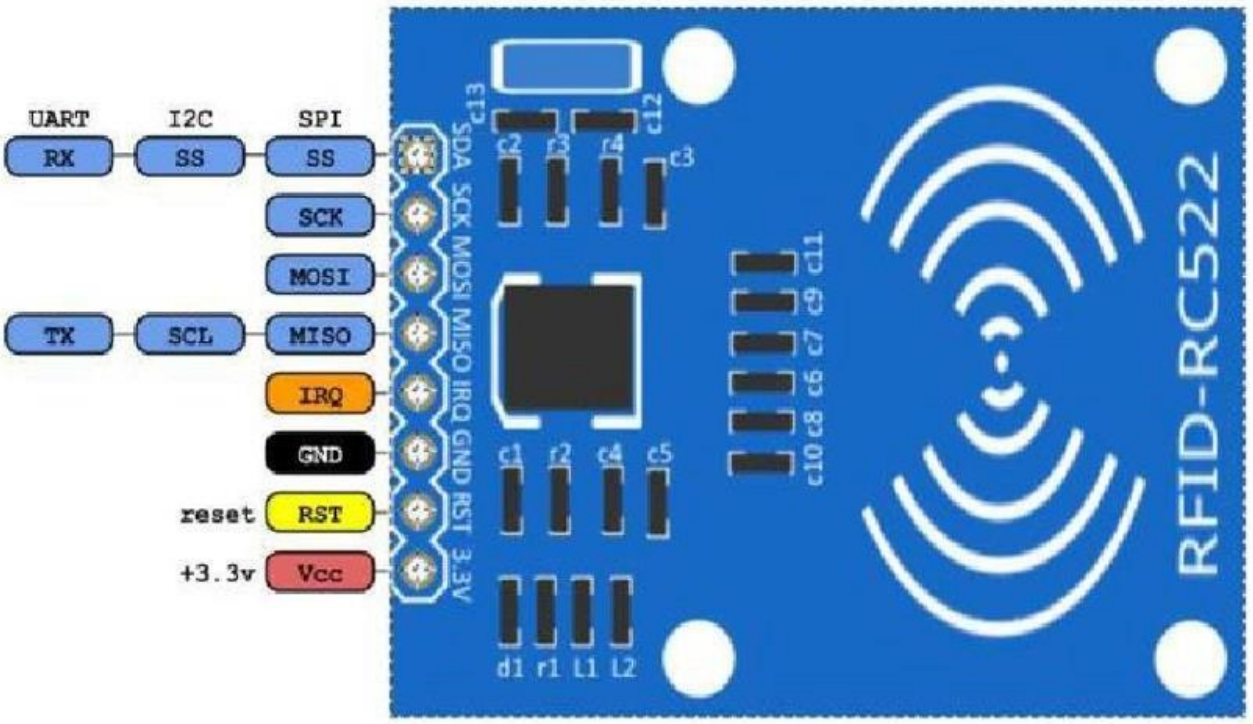


Рисунок 2.2 – Зчитувач RC522

Контакти та сигнали RFID RC522:

– VCC - Харчування 3.3V;

– RST - Reset. Лінія скидання, вхід;

– GND - Ground. Земля;

– MISO - Master Input Slave Output - дані від провідного 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 та I2C, через які відбувається обмін даними з іншими приладами. На платі модуля RFID RC522 установкою логічних рівнів спеціальних висновках мікросхеми обраний інтерфейс SPI. З одним Arduino може працювати кілька приладів, підключених до шини SPI.

Підключення модуля RC533 до Arduino Uno здійснюється відповідно до таблиці 2.1. Для нормальної роботи представленого вище модуля вихід IRQ не підключається Arduino.

Таблиця 2.1 – Підключення RC522 до Arduino Uno

MFRC522	Arduino Uno
RST	9
SDA	10
MOSI	11
MISO	12
SCK	13
3.3V	3.3V
GND	GND

В комплекті з даним модулем входить біла пластикова карта Mifare Classic 1K або мітка у вигляді брелока, зображена на рисунку 2.3.

Рисунок 2.3 – RFID-мітка Mifare 1K

Всередині неї знаходяться антена та мікросхема Mifare S50, що містить пам'ять та радіочастину. Розмір пам'яті 1 кг, тип EEPROM. Вона поділена на 16 секторів, що складаються з 4 розділів. У кожному розділі три інформаційні частини та одна для ключів. Усередині однієї частини є 16 байт пам'яті. Термін зберігання даних 10 років, кількість циклів перезапису 100 000.

Унікальність картки Mifare забезпечується присвоєнням виробником номера, що використовується як ідентифікаційний код. Для захисту даних у мікросхемі картки використано апаратне шифрування. Під час роботи дані з картки надходять на зчитувач тільки після взаємної ідентифікації коду, записаного в сектор пам'яті картки та зберігається у зчитувачі.

Для роботи в середовищі розробки Arduino можна скористатися різними сторонніми бібліотеками, розробленими для того, щоб спростити роботу зі зчитувачем RC522. Для запису та читання карти необхідно знати про її унікальний номер, необхідний для роботи системи радіочастотної ідентифікації. Для цього потрібно завантажити програму зі списку прикладів бібліотеки RFID під назвою CardInfo, підключити RC522 до Arduino і запустити програму в середовищі розробки Arduino IDE. При знаходженні робочої мітки в зоні дії RFID – зчитувача на моніторі порту з'явиться інформація про карту, яка представлена на рисунку 2.4.

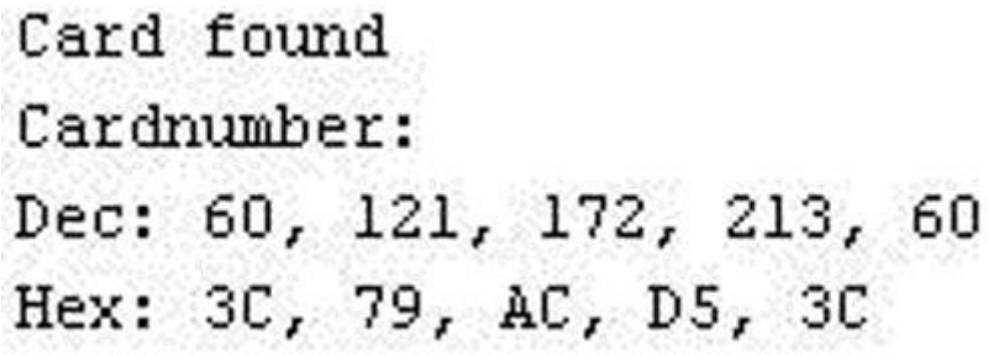


Рисунок 2.4 – Ідентифікаційні номери RFID – міток

Програма виводить ряд чисел: 60, 121, 172, 213, 60. Необхідно записати їх у зворотному порядку. Перше число виключається (контрольна сума), яке спочатку було останнім, а цифри, що залишилися, переводяться в шістнадцятковий код. Потім вони записуються в тому самому порядку, але без пробілів. Отримане велике число необхідно перевести в десятковий код, внаслідок чого вийде ідентифікаційний номер картки. З його допомогою вже можна проводити різні маніпуляції та складати різні програми, наприклад системи контролю доступу до приміщення.

Для того щоб вводити кодову комбінацію, можна скористатися спеціально сконструйованою для роботи з мікроконтролерами матричною клавіатурою, що складається з 16 кнопок, розташованих у 4 рядах і 4 стовпцях, внутрішня схема якої зображена на рисунку 2.5.

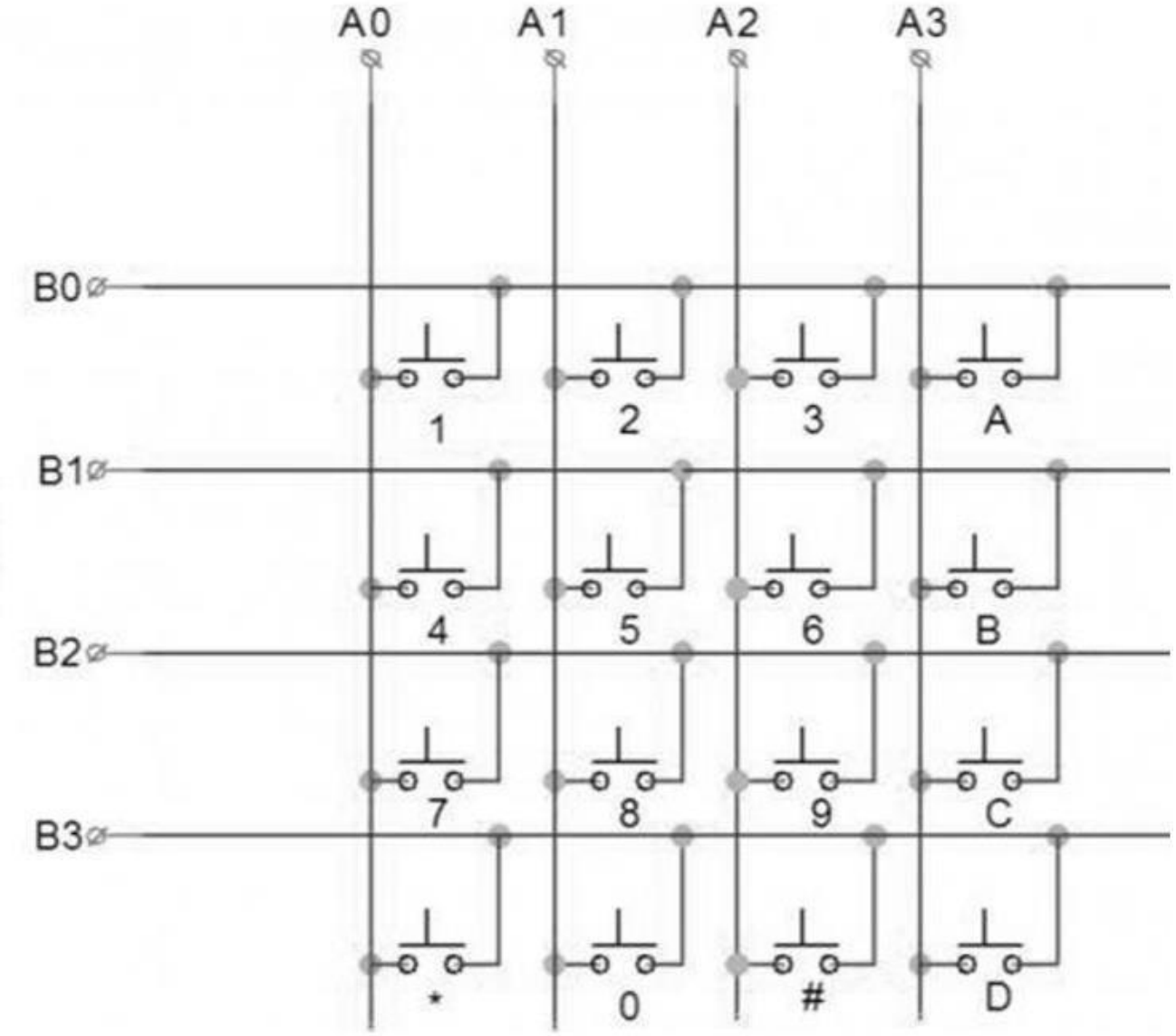


Рисунок 2.5 – Схема матричної клавіатури 4x4

Матричні клавіатури для мікроконтролерів досить різноманітні у своїй побудові. Крім 16-кнопкових клавіатур існують рішення з 12 або 4 кнопками, з мембранною підкладкою або з простими кнопками. Для вирішення поставлених завдань скористаємося типовим рішенням у вигляді матричної клавіатури з 16 кнопок, види якої представлені на рисунку 2.6.

Щоб підключити матричну клавіатуру до Arduino, від плати виведено 8 контактів, які підключаються через з'єднувальні дроти до цифрових входів мікроконтролера.

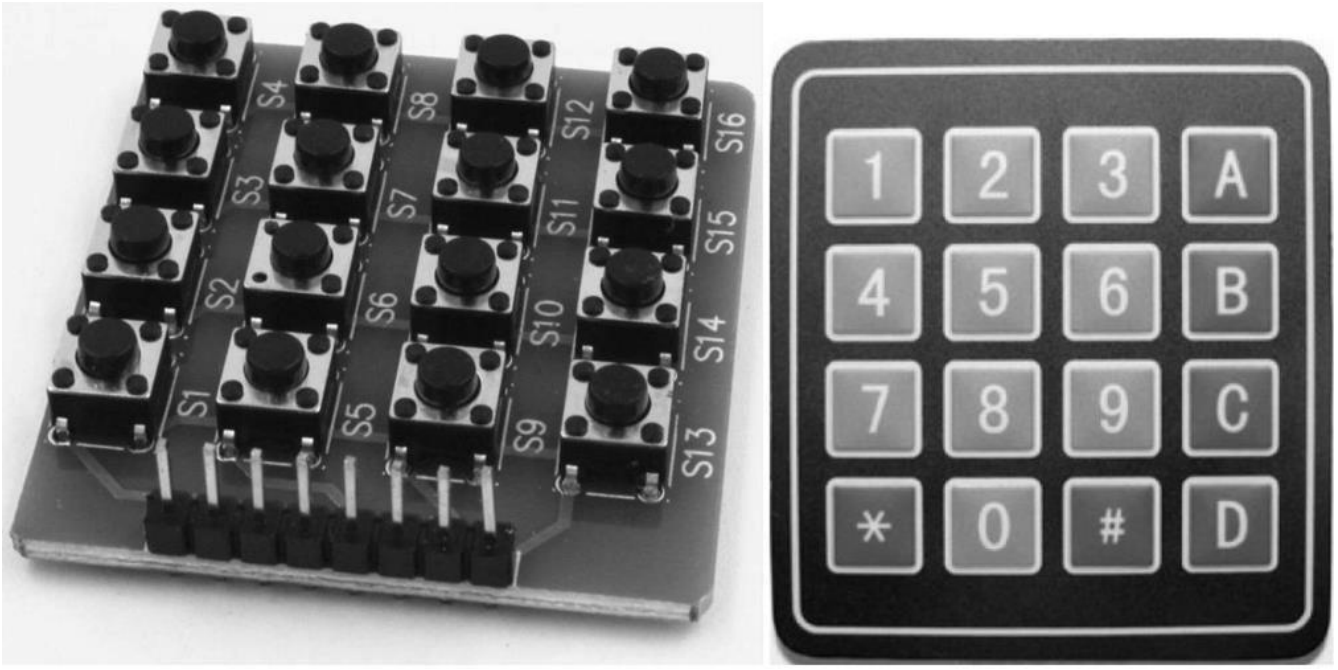


Рисунок 2.6 – Кнопкова та мембранна клавіатури 4x4

В пристрої багатьох електронних замків може бути присутнім елемент, який відтворює звук. Для цього підійде п'єзокерамічний випромінювач звуку (п'єзодинамік), який може відтворити звук на основі п'єзоелектричного ефекту. П'єзодинамік, зображений на рисунку 2.7, складається з металевої пластини, з нанесеної на ній п'єзоелектричної кераміки, що має струмопровідне напилення. Пластина та напилення є контактами п'єзовипромінювача, полярність яких – плюс та мінус. Якщо до контактів додати напруга, під дією зворотного п'єзоелектричного ефекту випромінювач почне відтворювати звук, а якщо механічно впливати на п'єзоелемент, то на його контактах з'явиться напруга. Щоб підключити п'єзодинамік до мікроконтролера, контакт, позначений знаком «+», підключається до будь-якого цифрового входу, а мінусовий контакт до виходу GND.

Рисунок 2.7 – Простий п'єзокерамічний випромінювач

Зручним способом відображення різної інформації, необхідної під час роботи пристрою контролю доступу є рідкокристалічний дисплей. Виберемо найпростіший в управлінні символьний LCD1602 дисплей на основі контролера HD44870, зображений на рисунку 2.8. На цьому рисунку також зображено плату послідовного I2C-інтерфейсу на основі мікросхеми PCF8574AT, за допомогою якої можна підключати дисплей до мікроконтролера через 4 дроти. Підключення відбувається дуже просто: SDA та SCL входи підключаються до входів SDA та SCL мікроконтролера, VCC та GND відповідно до +5В та GND мікроконтролера.

Рисунок 2.8 – Дисплей LCD1602 та плата підключення I2C-інтерфейсу

Для наочності використовуємо модуль RGB-світлодіода KY-016, представлений рисунку 2.9. У цьому модулі висновки, що відповідають за передачу кольору, вже підключені через резистори номіналом 220 Ом, тому немає потреби в окремих резисторах, щоб захистити світлодіод від виходу з ладу. Висновки R, G, B з'єднуються з цифровими входами мікроконтролера, а виведення "-" до входу GND.

Рисунок 2.9 – RGB-світлодіод KY-016

Для комутації різних приладів використовується спеціальний одноканальний модуль реле для мікроконтролерів, зображений рисунку 2.10, а принципова схема пристрою – рисунку 2.11.

Рисунок 2.10 – Схематичне зображення реле (вид зверху)

До складу реле входять: резистори номіналом 1 кОм (R1, R2), підтягуючий резистор R3 на 10 ком, pnp транзистор(VT1), зворотний діод(VD2) і реле (K1). VD1 (червоний світлодіод) – індикація подачі живлення на модуль, спалах VD3 (зелений світлодіод) свідчить про замикання реле. Контакти реле показано рисунку 2.12.

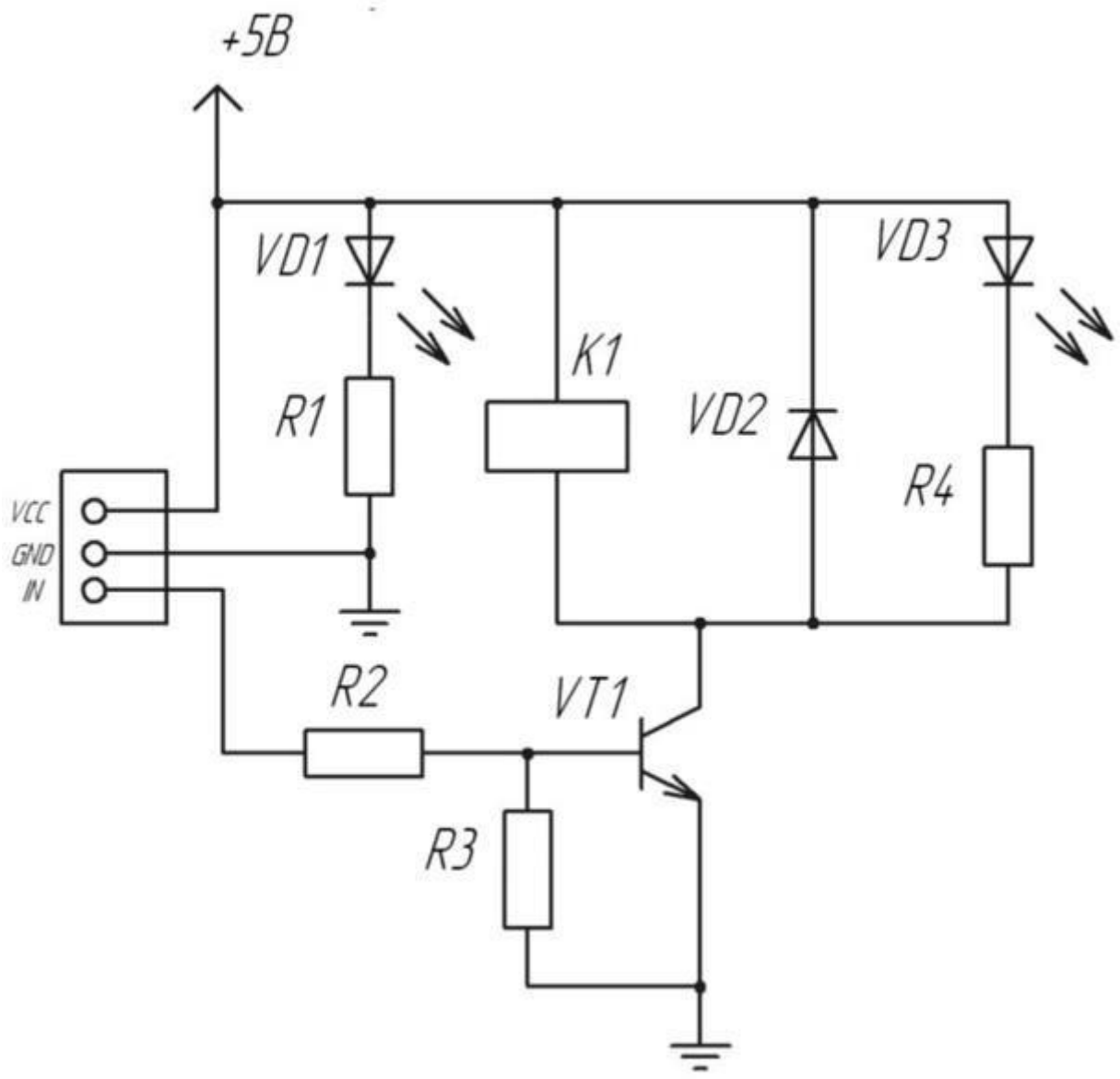


Рисунок 2.11 – Принципова схема модуля реле

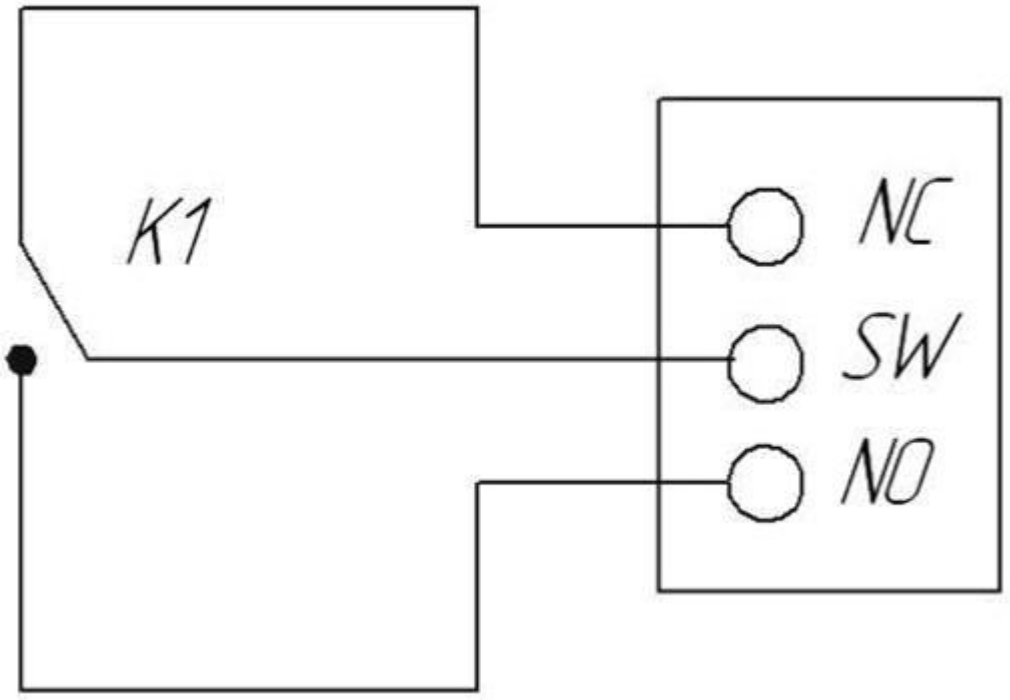


Рисунок 2.12 – Контакти одноканального реле

При включенні висновки перебувають у високоомному стані, транзистор не відкритий. Так як транзистор типу pnp, то для його відкриття потрібно подати на базу мінус. Для цього необхідно використовувати функцію digitalWrite(pin, LOW). Реле спрацьовує, коли транзистор відкритий і через керуючий ланцюг тече струм. Для відключення реле слід закрити транзистор, подавши з урахуванням плюс з допомогою функції digitalWrite(pin, HIGH). Контакти реле NC – нормально замкнутий, SW – контакт перемикання, NO – нормально розімкнений.

2.2 Розробка конструкції пристрою

Пристрої контролю доступу відіграють важливу роль у забезпеченні безпеки приміщень, надаючи контроль і обмеження доступу тільки авторизованим особам. Розробка ефективної конструкції такого пристрою є ключовим аспектом для забезпечення його надійності та функціональності. У цій статті ми розглянемо

процес розроблення конструкції пристрою контролю доступу в приміщення.

Перед початком розробки необхідно провести аналіз вимог і сценаріїв використання пристрою контролю доступу. Визначте, які функції та можливості мають бути включені в пристрій. Наприклад, це може бути використання різних методів ідентифікації, як-от біометричні дані або зчитування карток, а також функції аудиту доступу та системи управління.

Визначте відповідну фізичну конструкцію для пристрою контролю доступу. Це може бути автономний пристрій, що вбудовується в стіну або ворота, або навіть мобільний пристрій для тимчасового контролю доступу. Врахуйте вимоги до розмірів, зовнішнього вигляду і монтажних можливостей для обраного типу приміщення.

Виберіть необхідні компоненти для реалізації функцій пристрою контролю доступу. Це може включати в себе зчитувачі карт, біометричні сканери, електромагнітні замки, датчики руху та інші елементи. Врахуйте сумісність компонентів і можливість їхньої інтеграції з обраною платформою або протоколом зв'язку.

На цьому етапі необхідно розробити електричну схему та друковану плату для керування компонентами пристрою контролю доступу. Плануйте підключення компонентів, визначайте необхідні інтерфейси та врахуйте вимоги до живлення. Розробіть друковану плату з урахуванням розмірів і обмежень монтажного місця.

Розробіть програмне забезпечення для керування пристроєм контролю доступу. Використовуйте відповідне середовище розробки та мову програмування, сумісні з обраним контролером або платформою. Налаштуйте параметри роботи, наприклад, час затримки або порогові значення ідентифікації.

Проведіть тестування пристрою контролю доступу, щоб переконатися в його працездатності та відповідності вимогам. Ідентифікуйте та усуньте можливі проблеми або невідповідності. Оптимізуйте роботу пристрою, наприклад, покращуючи швидкість ідентифікації або керування доступом.

Після успішного тестування пристрою контролю доступу необхідно провести його інтеграцію та монтаж в обраному приміщенні. Врахуйте вимоги до

безпеки, естетики та зручності використання під час вибору місця встановлення. Встановіть пристрій згідно з інструкціями та рекомендаціями виробника.

Розробка конструкції пристрою контролю доступу в приміщення вимагає ретельного аналізу вимог, вибору компонентів, розробки електричної схеми та програмного забезпечення, а також тестування та інтеграції. Якісний пристрій контролю доступу забезпечує безпеку приміщень і зручність керування доступом.

Для представлення конструкції пристрою контролю доступу скористаємося дуже зручною програмою візуалізації всіх етапів роботи з Arduino. За допомогою програми Autodesk Circuits з електронного ресурсу circuits.io, що розповсюджується на безкоштовній основі та не потребує встановлення на персональний комп'ютер, можна моделювати різні рішення, які будуть показані у зручній для користувача формі макета або схеми підключення.

Промоделюємо модулі з допомогою Autodesk Circuits, використовувані разом із мікроконтролером Arduino, і виведемо їх схеми підключення до Arduino Uno R3, показані рисунки 2.13 – 2.16.

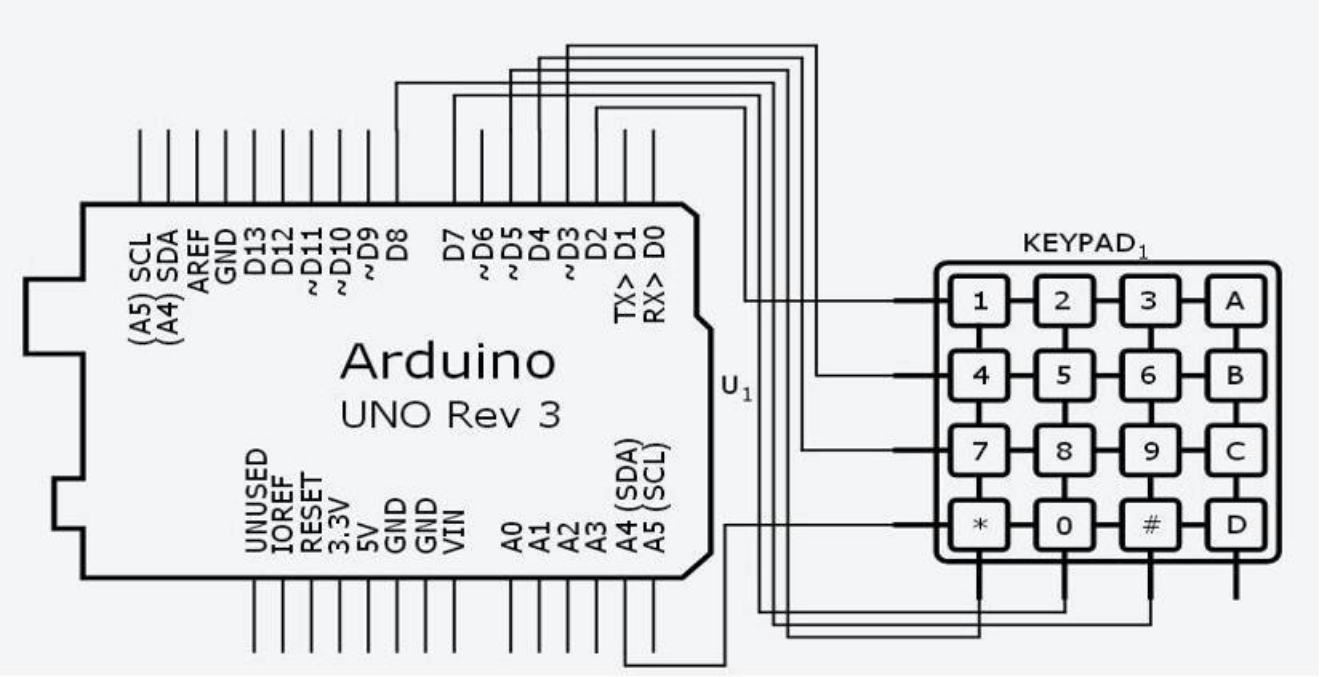


Рисунок 2.13 – Схема підключення матричної клавіатури

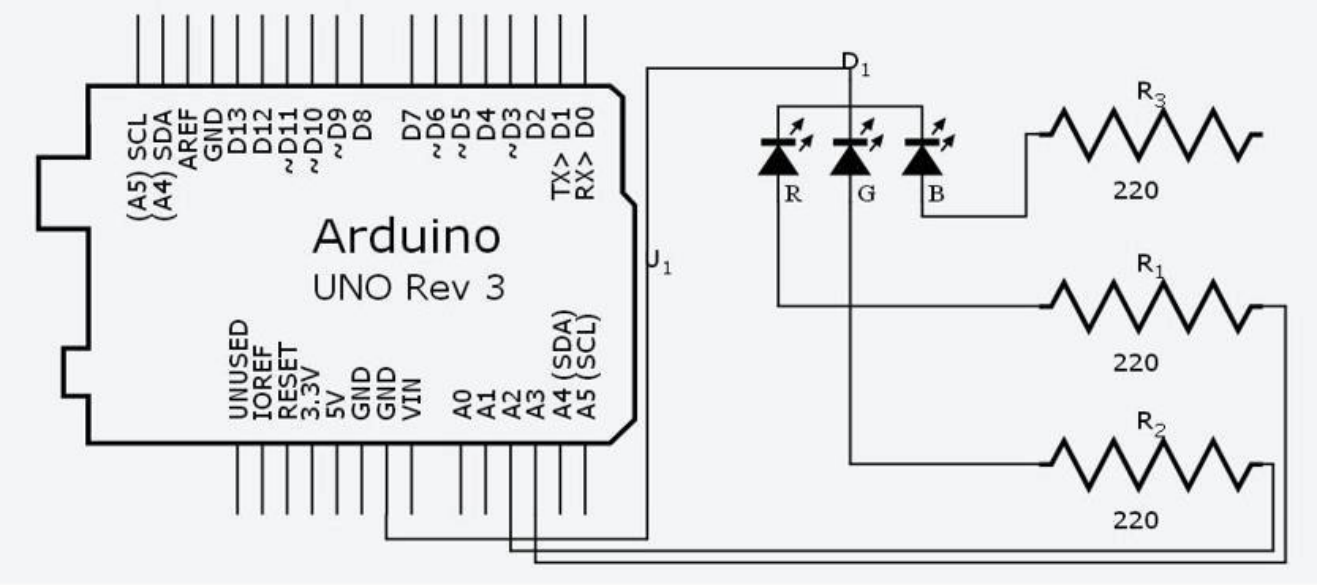


Рисунок 2.14 – Схема підключення RGB – світлодіода

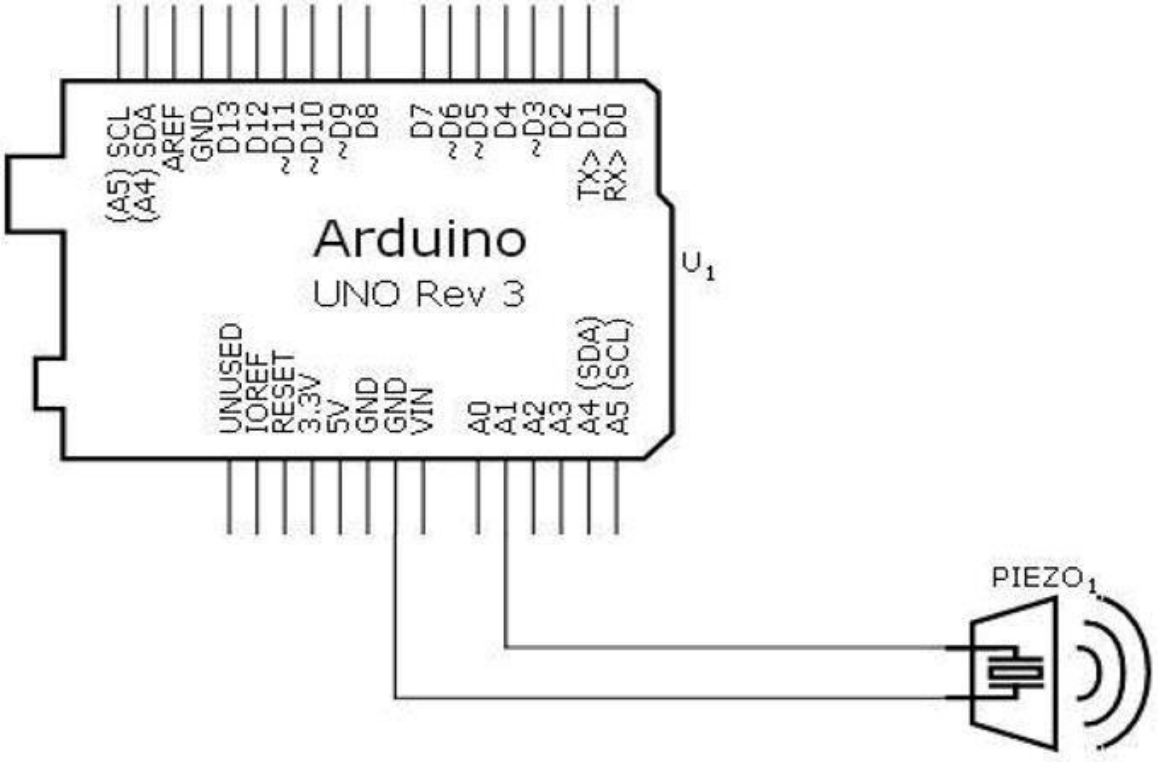


Рисунок 2.15 – Схема підключення п'єзодинаміка

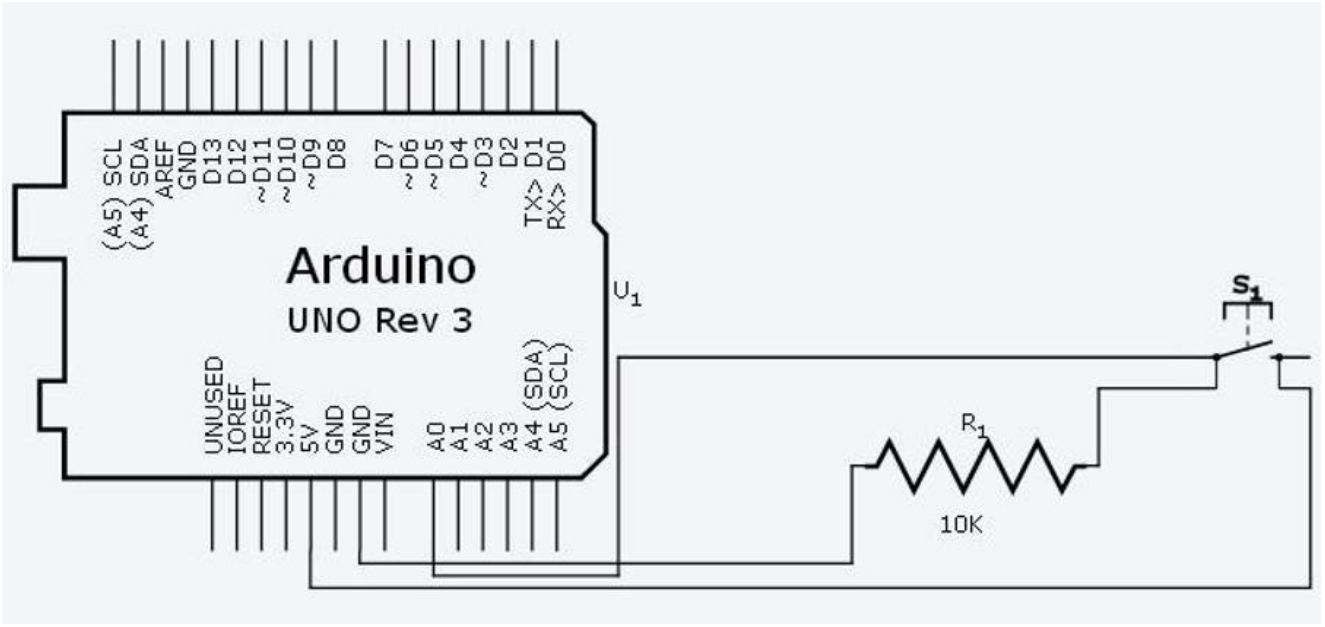


Рисунок 2.16 – Схема підключення кнопки скидання EEPROM

Незважаючи на всі переваги представленої вище програми, у неї є істотний недолік - дуже обмежена кількість елементів, зякими можна працювати. Тому подальше моделювання системи проводилося у програмному середовищі розробки Fritzing, спеціально розробленого для моделювання схем Arduino. Схеми підключень, створені у програмному середовищі Fritzing, зображені рисунки 2.17 – 2.19.

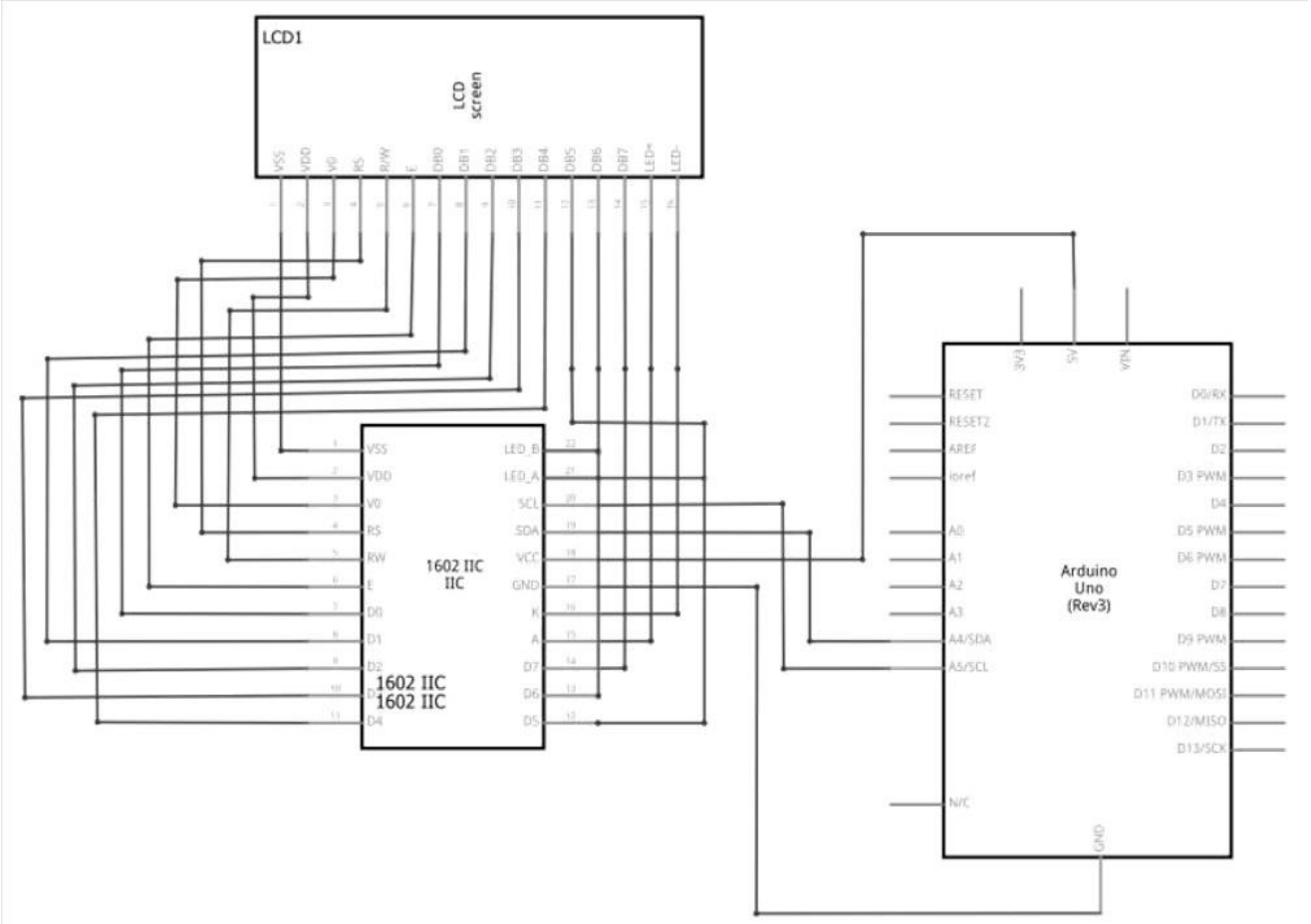


Рисунок 2.17 – Схема підключення LCD1602 за допомогою інтерфейсу I2C

Незважаючи на всі переваги представленої вище програми, у неї є істотний недолік - дуже обмежена кількість елементів, зякими можна працювати. Тому подальше моделювання системи проводилося у програмному середовищі розробки Fritzing, спеціально розробленого для моделювання схем Arduino. Схеми підключень, створені у програмному середовищі Fritzing, зображені рисунки 2.17.

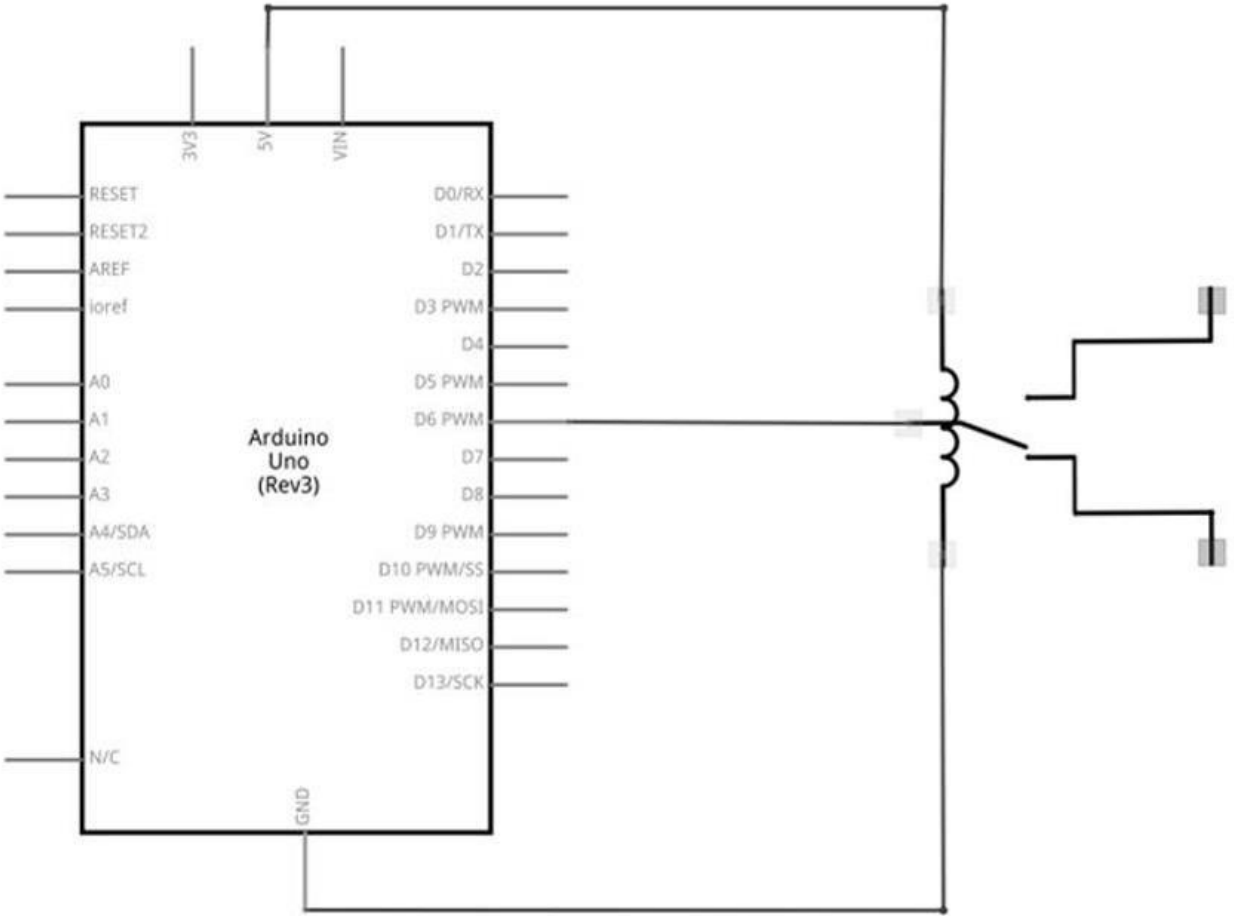


Рисунок 2.18 – Схема підключення одноканального реле

Незважаючи на всі переваги представленої вище програми, у неї є істотний недолік - дуже обмежена кількість елементів, зякими можна працювати. Тому подальше моделювання системи проводилося у програмному середовищі розробки Fritzing, спеціально розробленого для моделювання схем Arduino. Схеми підключень, створені у програмному середовищі Fritzing, зображені рисунки 2.19.

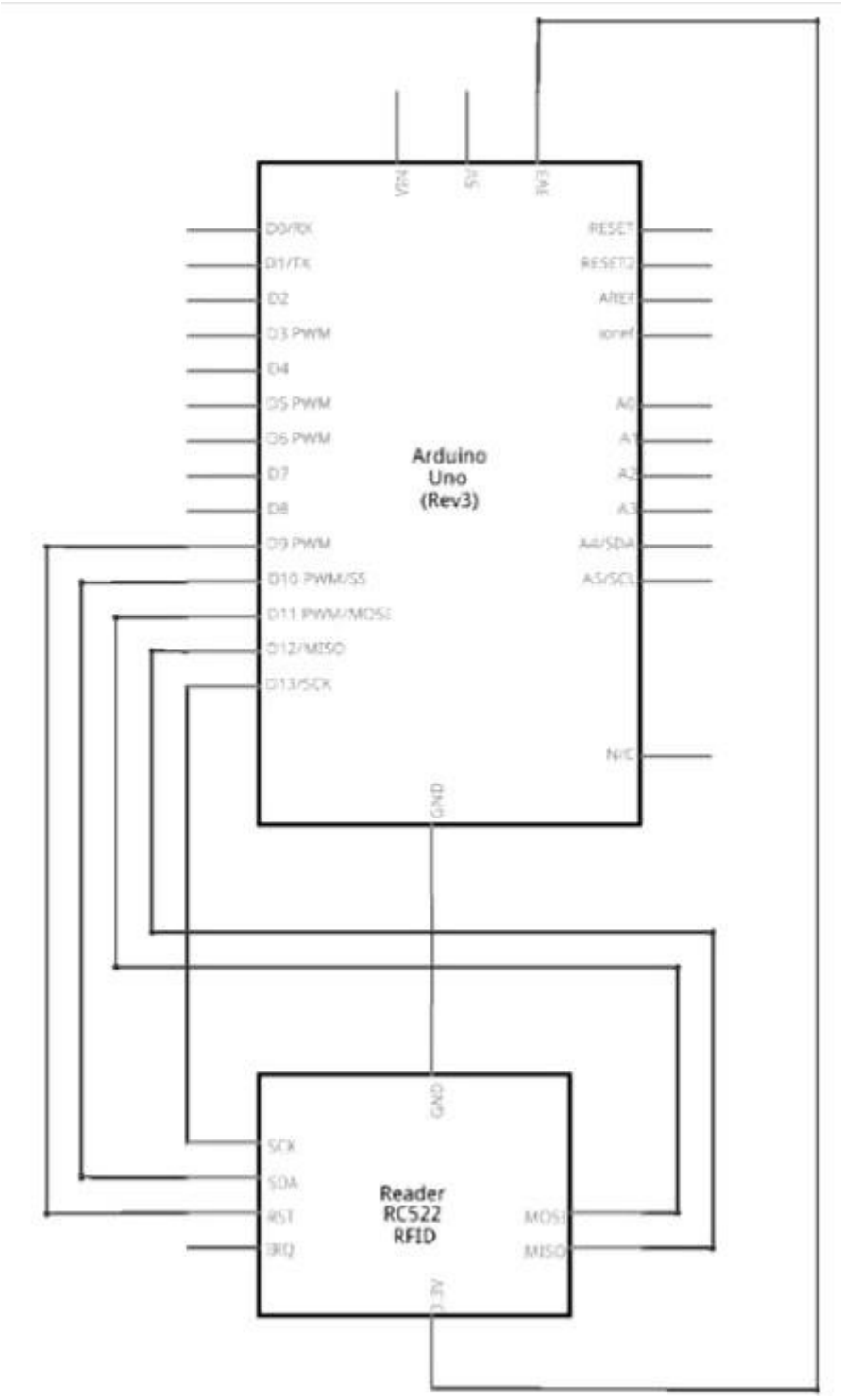


Рисунок 2.19 – Схема підключення RFID-зчитувача RC522

У результаті поєднали всі модулі однією схемою підключення, представленої рисунку 2.20.

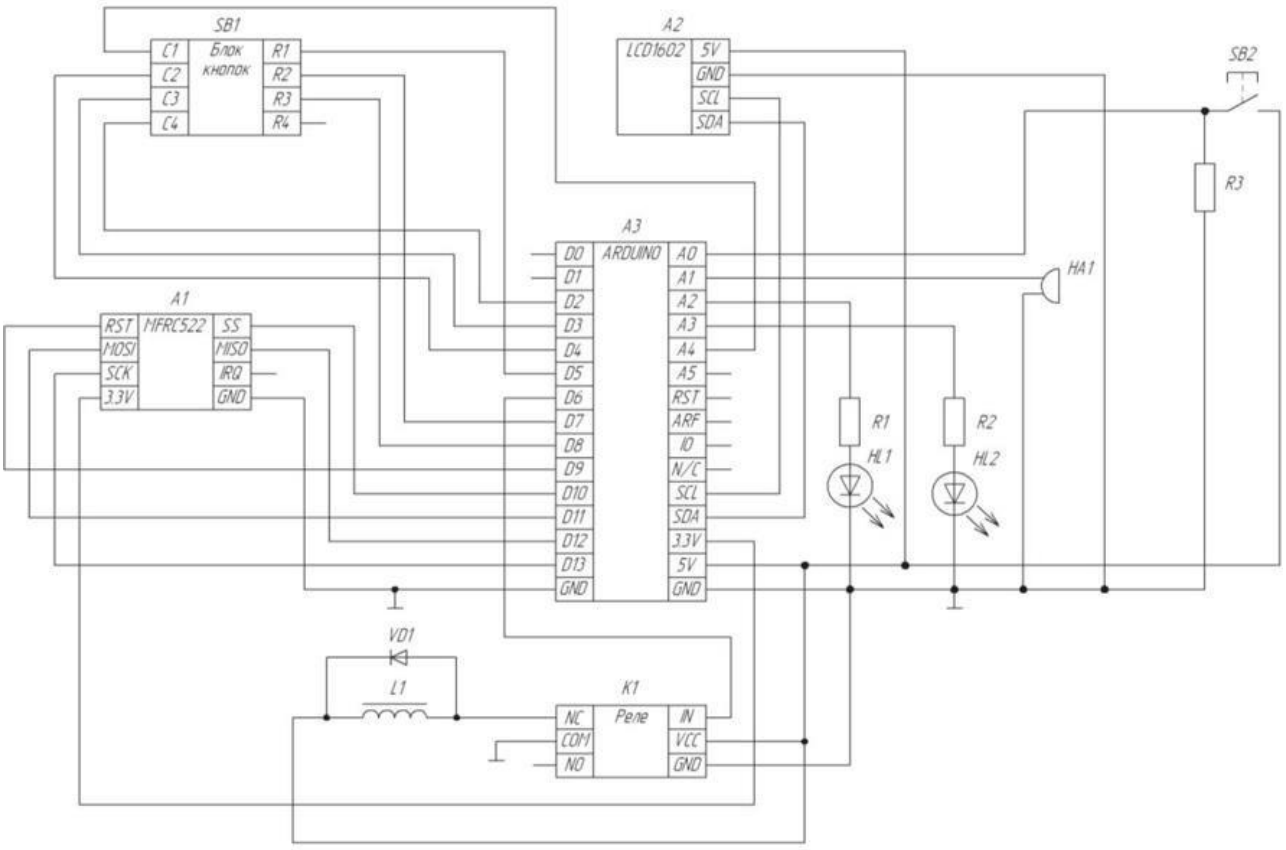


Рисунок 2.20 – Схема підключення всіх необхідних компонентів для контролю доступу до мікроконтролера

2.3 Розробка програмної частини пристрою

Для того, щоб почати розробляти програму необхідно спочатку задатися базовим алгоритмом роботи пристрою, необхідним для розуміння процесів виконання різних ситуацій (рис. 2.19).

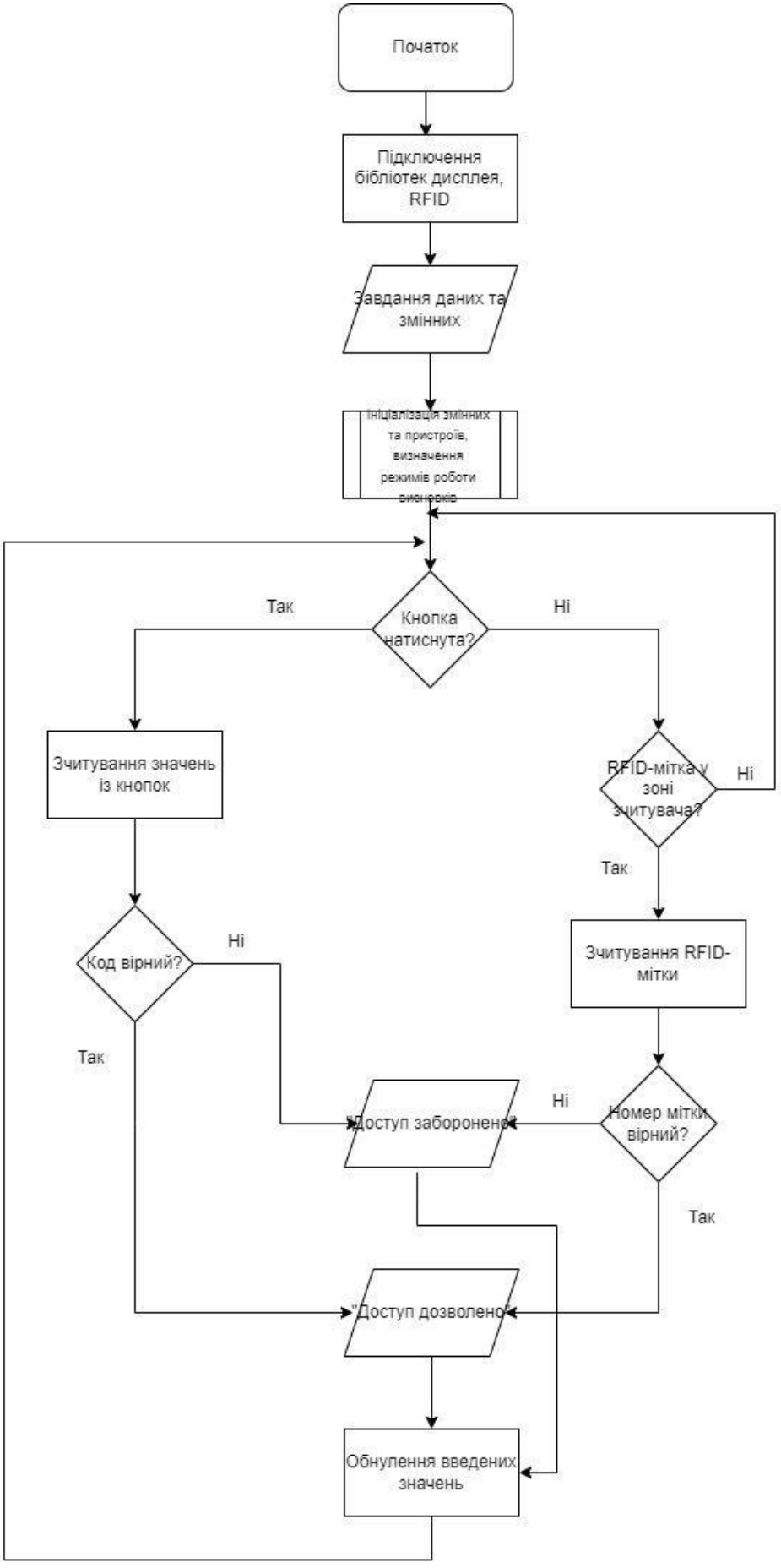


Рисунок 2.19 – Базовий алгоритм роботи пристрою контролю доступу

Для написання програми скористаємося спеціально розробленим середовищем розробки Arduino IDE, яке розповсюджується у вільному доступі в мережі Інтернет і яке можна знайти на офіційному сайті розробника www.arduino.cc. В результаті, після тривалого процесу розробки шляхом спроб і помилок було складено програму, за допомогою якої можна реалізувати основний функціонал роботи електронного замку на основі Arduino Uno R3. Для правильної роботи програми необхідно заздалегідь встановити кілька сторонніх бібліотек або через програму Arduino - Скетч - Підключити бібліотеку - Управляти бібліотеками - Пошук, або через ресурси мережі Інтернет, встановивши її по шляху Arduino - Скетч» - «Підключити бібліотеку» - «Додати .ZIP бібліотеку». Необхідні сторонні бібліотеки для роботи з програмою: "Keypad", "Password", "MFRC522", "LCD_1602_RUS", "Bounce2". Програма, що реалізує потенціал пристрою контролю доступу до приміщення, наведена в додатку А.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Виготовлення системи

Для підключення різних модулів до мікроконтролера Arduino існує спеціальна макетна плата, на якій зручно розташовувати різні елементи схеми, а також з'єднувати їх проводами між собою та мікроконтролером. Зрештою, вийшла схема, представлена рисунку 3.1. Всі елементи з'єднані з мікроконтролером згідно з принциповими схемами, які були розроблені у спеціалізованих програмах моделювання Autodesk Circuits та Fritzing.

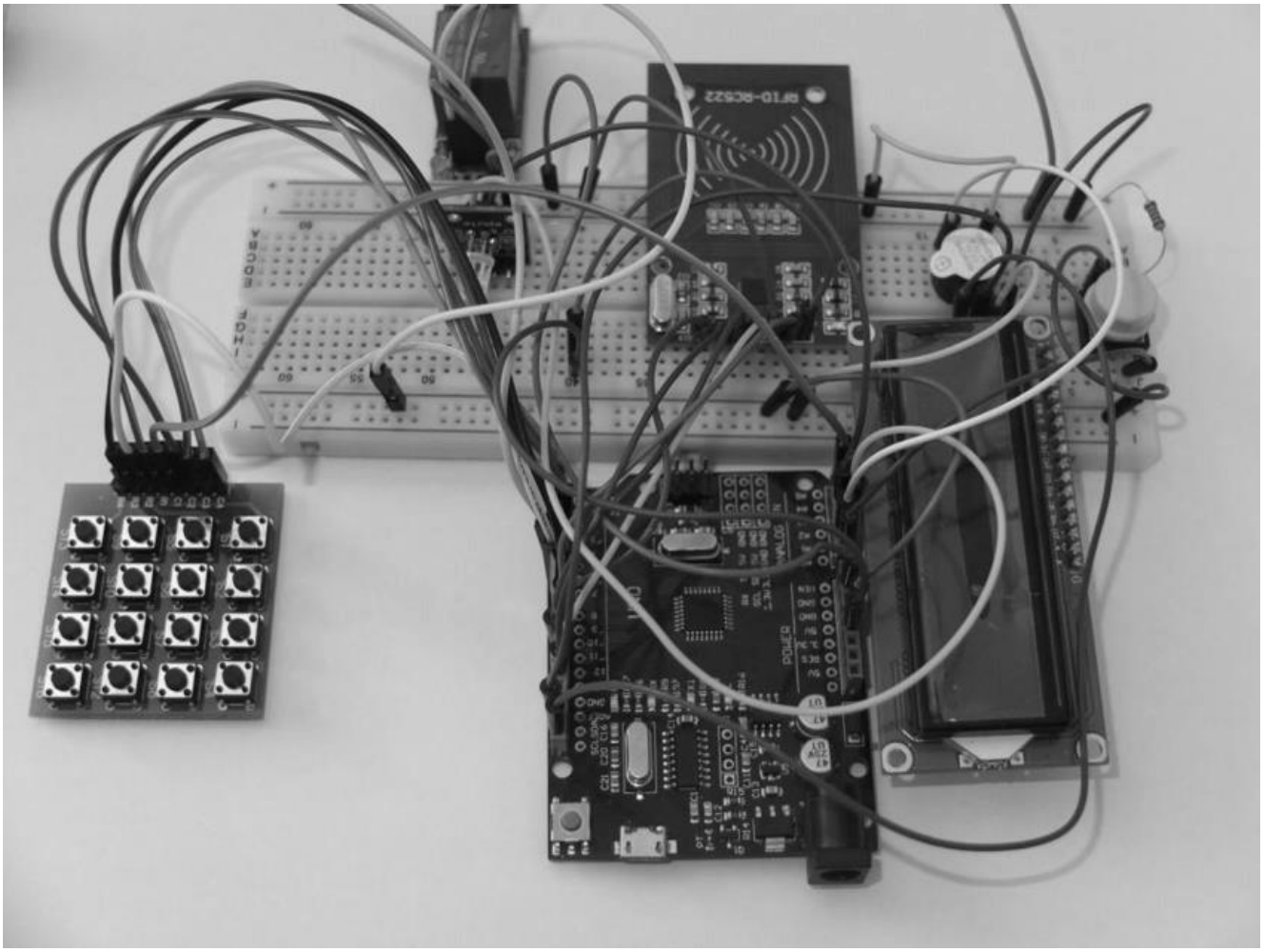


Рисунок 3.1 – Повна схема контролю доступу

Розглянемо підключення RFID-зчитувача RC522 крупним планом, представленим рисунку 3.2, і світлодіода, зображеного рисунку 3.3.

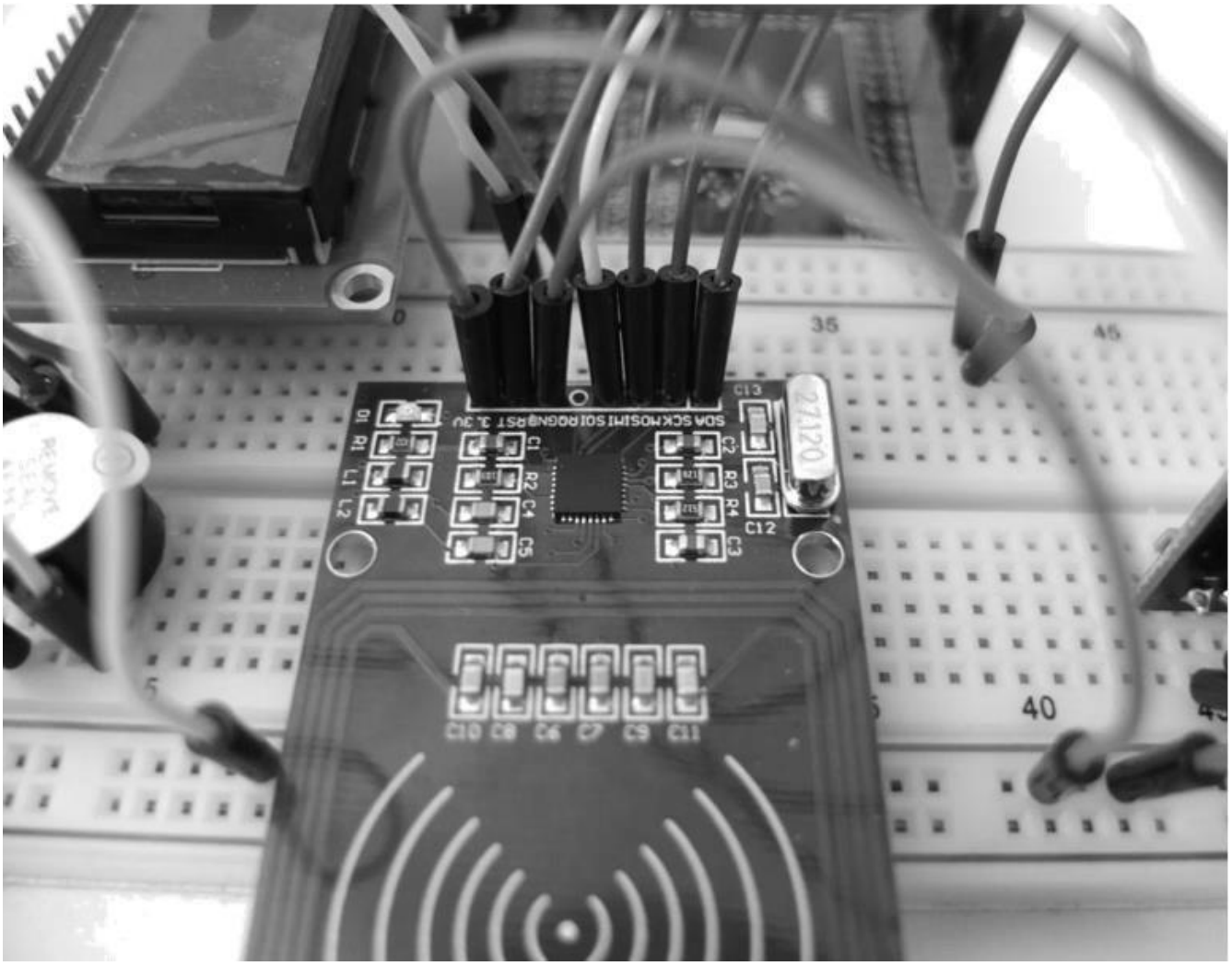


Рисунок 3.2 – З'єднання RC522 з мікроконтролером

Під час практичної роботи та створення моделі електронного пристрою контролю доступу виникли деякі проблеми, а також нові ідеї щодо модернізації програмної частини. Зокрема, було проведено заміну робочої бібліотеки RFID з Rfid.h на MFRC522, у якій функціонал перевершував попередню бібліотеку. Потім була додана окрема функція squeaker(), за допомогою якої було простіше та зручніше налаштовувати п'єзодинамік.

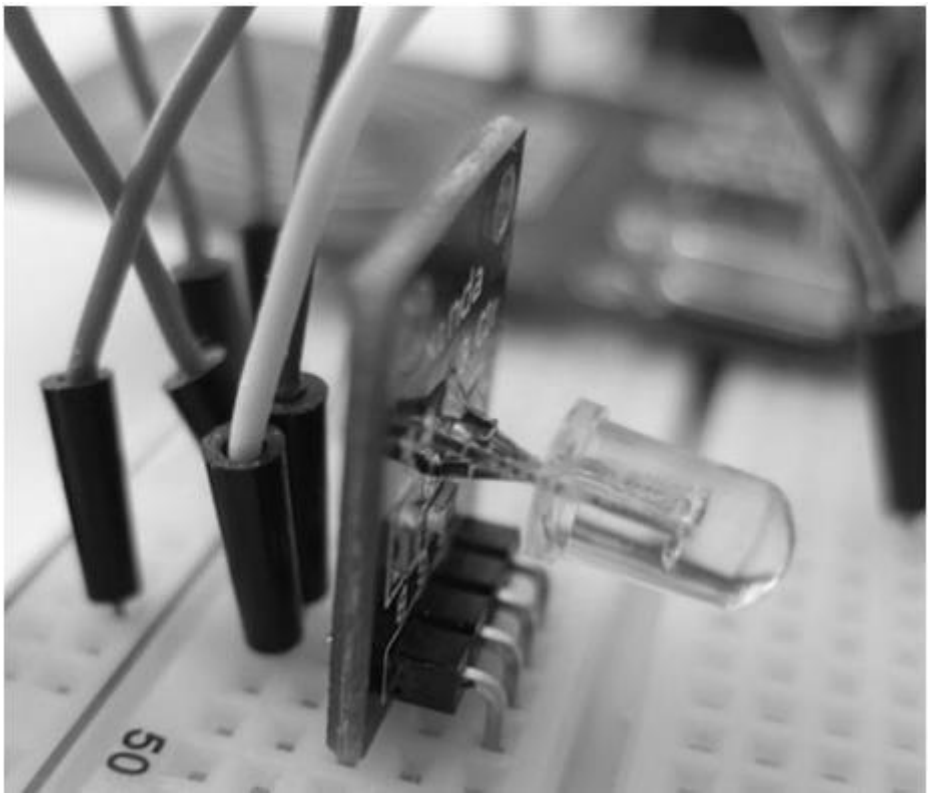


Рисунок 3.3 – З'єднання RGB-світлодіода з мікроконтролером

3.2 Перевірка та налагодження прграмної частини пристрою

Під час практичної роботи та створення моделі електронного пристрою контролю доступу виникли деякі проблеми, а також нові ідеї щодо модернізації програмної частини. Зокрема, було проведено заміну робочої бібліотеки RFID з Rfid.h на MFRC522, у якій функціонал перевершував попередню бібліотеку. Потім була додана окрема функція squeaker(), за допомогою якої було простіше та зручніше налаштовувати п'єзодинамік. На платі мікроконтролера Arduino UNO R3 є два додаткові входи SDA і SCL, що дозволяють працювати за протоколом I2C з різними пристроями, у нашому випадку, з дисплеєм LCD1602. Зміни відбулися і з інформацією, що відображалася на дисплеї. Спочатку всі символи під час роботи пристрою були записані латинськими літерами. Вирішити це питання допомогла стороння бібліотека LCD_1602_RUS.h, що дозволяє використовувати кирилицю.

Зрештою, найголовнішим рішенням було додавання програмної роботи з енергонезалежною пам'яттю EEPROM мікроконтролера, внаслідок чого ідентифікаційні номери RFID-міток, а потім і кодова комбінація, що вводиться за допомогою матричної клавіатури. Вони почали записуватися в цю незалежну пам'ять і при роботі програми виводитися в послідовний порт персонального комп'ютера, а не задаватися в самій програмі, як було раніше. Разом з EEPROM була додана спеціальна кнопка скидання пам'яті мікроконтролера, яка заміняла всі дані в EEPROM на нульові значення, крім значень, що використовуються в паролі.

Тепер слід перевірити основні етапи роботи пристрою контролю доступу. На рисунку 3.4 показано інформацію, що виводиться в послідовний порт комп'ютера. Інформацію з послідовного порту можна відобразити на екрані монітора, зайшовши в середу розробки Arduino IDE, в рядку «Інструменти» - «Порт» вказати номер COM порту якому підключений мікроконтролер, а потім у рядку «Інструменти» - «Монітор порту», що дозволить вивести налагоджувальну інформацію на екран монітора.

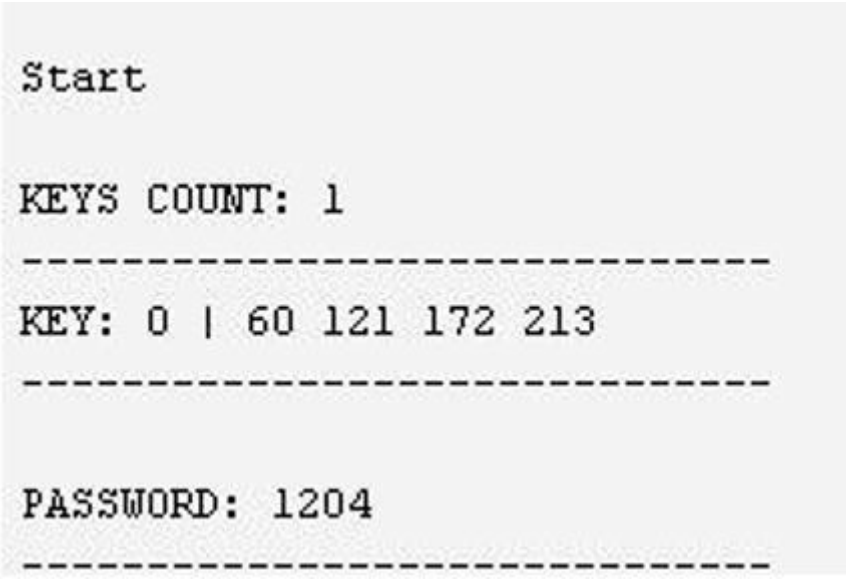


Рисунок 3.4 – Відлагоджувальна інформація, що вказує кількість доступних системі ключів, їх ідентифікаційний номер, а також пароль

При запуску програми з чистою EEPROM пам'яттю, або коли вона була очищена за допомогою спеціальної кнопки скидання, на екрані з'явиться

інформація, представлена на рисунку 3.5. В цьому випадку контакти реле будуть замкнені, таким чином моделюючи відкритий стан, програма запросить додати RFID-ключ до зчитувача, який згодом стане майстер-ключом.

```
Memory cleaning is completed


Start

The master key is not in memory. The first presentation to the key will be the master!
```

Рисунок 3.5 – Відлагоджувальна інформація, що вказує на те, що в пам'яті мікроконтролера відсутні RFID-мітки.

При установці майстер-ключа програма запише його ідентифікаційний номер і вкаже на його належність (рисунок 3.6).

```
UID: 60 121 172 213
master key is created
```

Рисунок 3.6 – Створення майстер-ключа

Майстер-ключ дозволяє вносити на згадку мікроконтролера нові ключі при його утриманні у зчитувача, а також виходити з цього режиму програмування за той же час утримання. У цей час реле замикає свої контакти і не розмикає доти, доки не відбудеться вихід з режиму програмування нових ключів (рисунок 3.7). Весь цей етап супроводжується звуковими сигналами та повідомленнями на дисплеї.

```
UID: 60 121 172 213
MASTER PROGRAMMING MODE ON

UID: 101 229 204 101
add key in eeprom

KEYS COUNT: 2
------------------------------
KEY: 0 | 60 121 172 213
KEY: 1 | 101 229 204 101
------------------------------

UID: 60 121 172 213
MASTER PROGRAMMING MODE OFF
```

Рисунок 3.7 – Режим програмування

Введення правильної кодової комбінації або одного із записаних у пам'яті ключів супроводжуються характерними звуковими сигналами, інформацією на дисплеї (рисунок 3.8), а також замиканням контактів реле та спрацьовуванням зеленого світлодіода протягом п'яти секунд, після система приходить у початкове положення. Неправильний код або невідомий ключ змушують систему реагувати негативно: пристрій недоступний на короткий проміжок часу, відтворюється характерний звуковий сигнал, на екрані дисплея з'являється повідомлення "Доступ заборонено", після чого система повертається до вихідної позиції.

Рисунок 3.8 – Інформація на дисплеї під час правильного вирішення

Кодова комбінація "0000" дозволяє входити в режим зміни пароля. Після цього дається три спроби введення попередньої комбінації, інакше виводиться повідомлення, зображене на рисунку 3.9.

```
Pass 7777
Attention!_3xWrong_Pass for change pas - cancel change pass
Access denied!
```

Рисунок 3.9 – Налагоджувальне повідомлення на спробу введення неправильного пароля втретє

При зміні пароля також неприпустимим є комбінація "0000", а також колишній пароль. У цьому випадку на дисплеї з'явиться повідомлення "Старий пароль", але програма дозволяє ввести новий. Коли нова комбінація успішно введена, спалахує зелений світлодіод, потім система переходить у початковий стан.

Інструкцію по роботі з пристроєм контролю доступу до приміщення викладено у додатку Б.

ВИСНОВКИ

В даній бакалаврській кваліфікаційній роботі здійснено розробку, створення робочої програми, виготовлення, налагодження та діючого макета пристрою контролю доступу до приміщення на мікроконтролерному управлінні.

Пристрій моделює основні функції контролю за різними датчиками та пристроями, що входять до периферії електронної системи.

Макет електронного замку дозволяє змоделювати основні функції завдання контролю доступу в різні приміщення за допомогою введення кодової комбінації через матричну клавіатуру, або радіочастотної мітки, що використовується як ключ-карти. Модуль реле і підключений до його контактів електромагніт симулюють відкритий або закритий стан дверей у приміщення. Під час втрати майстер-ключа передбачено скидання пам'яті мікроконтролера з метою вказівки нового майстер-ключа, який використовується для введення нових карт доступу.

Вся налагоджувальна інформація надходитиме на комп'ютер лише при підключенні до нього мікроконтролера через USB-інтерфейс. Таким чином, пристрій може працювати автономно через підключений мережевий блок живлення, але в цьому випадку стає недоступною інформація про налагодження.

Основною метою даної роботи було створення макета пристрою контролю доступу на мікроконтролерному управлінні. Отримана система дозволяє змоделювати основні функції, що доступні аналогічним механізмам без мікроконтролерів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 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. URL:https://www.researchgate.net/publication/275685766_RFID_Based_Security_and_Access_Control_System(Дата звернення: 4.04.2023).

2. A Digital Security System з системою Довжина блоку RFID Technology / GK Verma, P. Tripathi // International Journal of Computer Applications, Volume 5 – No.11, P.6-8, August 2010. URL:https://www.researchgate.net/publication/45602075_A_Digital_Security_System_with_Door_Lock_System_Using_RFID_Technology(Дата звернення: 4.04.2023).

3. A Review on Chipless RFID Tag Design / A.Hashemi, AHSarhaddi, H.Emami// Majlesi Journal of Electrical Engineering, vol.7, No.2, P.68-75, June 2013. URL:https://www.researchgate.net/publication/260190506_A_Review_on_Chipless_RFID_Tag_Design(Дата звернення: 4.04.2023).

4. Кодові замки на двері: види та їх особливості. URL:http://dveridoma.net/kodovye-zamki-na-dveri/(Дата звернення: 7.05.2023).

5. Електронний замок. URL:http://www.meanders.ru/kodovij_zamok.shtml(Дата звернення: 7.05.2017).

6. Arduino UNO R3: схема, інструкція. URL: https://www.syl.ru/article/203717/new_arduino-uno-r-shema-instruktsiya (дата звернення: 7.05.2023).

7. Electronic Wallet and Access Control Solution Based on RFID MiFare Cards / Stefan-Victor Lefter // Journal of Mobile, Embedded and Distributed Systems, vol.5, no.1, P29-35, 2013. URL:http://www.jmeds.eu/index.php/jmeds/article/view/Electronic_Wallet_and_Access_Control_Solution_Based_on%20%20_RFID_MiFare_Cards(Дата звернення:

4.04.2023).

8. Autodesk Circuits. URL:https://circuits.io(Дата звернення: 3.05.2023).
9. Fritzing.URL:http://fritzing.org/home/(Дата звернення: 3.05.2023).
10. What Everyone Must Know About Industry 4.0. Bernard Marr. Forbes 2016 URL:https://www.forbes.com/sites/bernardmarr/2016/06/20/what-everyone-must-know-about-industry-4-0/#b4c9ac7795f7(дата звернення 03.05.2023).
11. Інфрачервоний датчик Sharp GP2Y0A21YK0F 2019 URL:http://electromicro.com/market/datchiki_i_sensory/infrakrasnyj_datchik_gp2y0a21yk0f/(дата звернення 20.04.2023)
12. Round-robin (алгоритм) 2011 URL:https://wikipedia.org/wiki/Round-robin (дата звернення 21.05.2023)
13. Lora AT COMMANDGUIDE.REYAX TECHNOLOGY CO., LTD, 2018 року. Raspberry Pi Documentation. – Режим доступу:www.raspberrypi.org (дата звернення 21.04.2023)

Додаток А

Лістинг програми пристрою контролю доступу до приміщення

Початок програми:

```
#include <avr/wdt.h>
#include <Wire.h>
#include "LCD_1602_UA.h"
#include <SPI.h>
#include <MFRC522.h>
#include <Bounce2.h>
#include <EEPROM.h>
#include <Password.h>
#include <Keypad.h>
LCD_1602_UA lcd(0x27, 16, 2); // ініціалізація
дисплея, 16 стовпців, 2 рядки
//Визначення основних пінів, до яких підключаються різні
модулі:
#define PIN_RESET 14 // кнопка для скидання EEPROM
#define PIN_RELAY 6 //
підключення реле #define
PIN_TONE 15 // п'єзодинаміка
#define PIN_RST 9 // RFID RST
#define PIN_SS 10 // RFID SS
#define RED_LED 17 // червоний
світлодіод #define GREEN_LED 16
//зелений світлодіод
//Ініціалізація RFID-зчитувача:
MFRC522 mfrc522(PIN_SS, PIN_RST);
//Змінні, необхідні роботи зі списком RFID-ключів:
byte **keyss;
byte keys_count = EEPROM.read(0);
//Змінні необхідні для режиму програмування RFID-міток:
byte modeProgTime = 5; // Кількість секунд утримання
майстер ключа для входу чи виходу з режиму програмування
bool mode = false;
byte modeClean = 0;
unsigned long modeTimer = 0;
unsigned long resetTimer = 0;
//Управління замком:
unsigned long openTimer = 0;
```

```
//Захист кнопок від брязкоту:
Bounce key_reset = Bounce ();
Bounce key_open = Bounce ();
// Програмний reset:
void(* resetFunc) (void) = 0;
// Функція звукового оповіщення. Приймає параметри:
//звукових сигналів, частота в герцах, тривалість звуку,
пауза у //мілісекундах (не обов'язково):
void squeaker(byte count, unsigned int Hz, unsigned int
sleep = 0)
{
 for(int i=0; i<count; i++) {
 tone(PIN_TONE, Hz, тривалість);
 if(sleep > 0) delay(sleep);
 }
}
//Функція для зчитування EEPROM та складання списку RFID-
ключів.
//Перший байт у пам'яті містить кількість ключів. UID
ключа містить 4//байта. Максимум можна записати 254 ключі
(255-1 через те, що //EEPROM записується кодова
комбінація з 4 символів):
void keysRead() {
 //Виводимо кількість ключів:
 Serial.print(F("KEYS COUNT:"));
 Serial.println(keys_count);
 int eb = 4; // Запис кількості доступних ключів
 проводиться в keyss = (byte **) malloc (sizeof
 (byte *) * keys_count);
 //Читаємо список ключів з EEPROM:
 Serial.println(F("----------------------------
 ---------")); for(byte i=0;
 i<keys_count; i++) {
 Serial.print(F("KEY:
 "));Serial.print(i);Serial.print(" |
 "); keyss[i] = (byte *) malloc (sizeof
 (byte) * 4);
 for(byte b=0; b<4; b++) {
 keyss[i][b] =
 EEPROM.read(++eb);
 Serial.print(keyss[i][b]);
 if(b < 3) Serial.print(F(" "));
 }
 Serial.println();
 }
```

```
 Serial.println(F("--------------
----------------------"));
 Serial.println();
}
//Функція виведення пароля, записаного з першого по п'ять
осередок пам'яті, т.к
//Змінні записуються в типі char, то потрібно перевести
код ASCII в
// десятковий. Для цифр від 0 до 9 це можна зробити
просто відніманням // отриманого результату числа 48:
void passRead() {
Serial.print(F("PASSWORD: "));
 Serial.print(EEPROM.read(1)-48);
 Serial.print(EEPROM.read(2)-48);
 Serial.print(EEPROM.read(3)-48);
 Serial.print(EEPROM.read(4)-48);
 Serial.println();
 Serial.println(F("----------------------------------------"));
 Serial.println(); }
//Функція виводить UID ключа і, за необхідності,
супровідне //повідомлення:
 void uidPrint(String text = "") {
 Serial.print(F("UID: "));
 for(byte i=0; i<mfrc522.uid.size; i++) {
 Serial.print(mfrc522.uid.uidByte[i]);
 if(i < mfrc522.uid.size - 1)
 Serial.print(F(" ")); }
 Serial.println();
 if(text.length() != 0)
Serial.println(text + "\r\n"); }
int presses=0; // кількість
натискань const byte ROWS = 4; //
кількість рядків const byte COLS =
4; // кількість стовпців
// Визначення символів матричної
клавіатури: char keys[ROWS][COLS] =
{
 {'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'*','0','#','D'}
};
//Цифрові входи, до яких підключається матрична
клавіатура:
byte rowPins[ROWS] = {2,3,4,18};
```

```
byte colPins[COLS] = {5,7,8,19};
//Ініціалізація матричної клавіатури:
Keypad keypad = Keypad( makeKeymap(keys), rowPins,
colPins, ROWS, COLS );
String pass; //Змінна, що вказує код за
замовчуванням String summ; //Змінна, що вказує
вже введений код
int wrong=0; //кількість помилкових вводів коду
(для блокування) int shetch=1; //кількість набраних
символів зміни пароля int change=0; //прапор
перевірки коду зміни пароля int dochange=0;
//прапор проводиться зміна пароля
//Функція, що запускається лише на початку роботи
мікроконтролера або //при апаратному скиданні:
void setup()
{
 //Налаштовуємо сторожовий
 таймер wdt_disable();
 // Delay (8000);
 wdt_enable(WDTO_8S);
//Запис вихідного пароля в незалежну пам'ять, необхідно
тільки //при початковій прошивці, щоб задати пароль:
//EEPROM.write(1, '1');
//EEPROM.write(2, '2');
//EEPROM.write(3, '0');
//EEPROM.write(4, '4');
//Запис символів у форматі char:
 pass="";
 pass = pass + char (EEPROM.read (1));
 pass = pass + char (EEPROM.read (2));
 pass = pass + char (EEPROM.read (3));
 pass = pass + char (EEPROM.read (4));
//Ці рядки тільки для налагодження, що виводиться в
COM-порт, до якого //підключений мікроконтролер:
Serial.println(char(EEPROM.read(1))); // перший знак
пароля в
 Serial.println(char(EEPROM.read(2))); // другий знак
 пароля
 Serial.println(char(EEPROM.read(3))); // третій знак
 пароля
 Serial.println(char(EEPROM.read(4))); // Четвертий знак
 пароля
 Serial.print("Pass"); Serial.println(pass); //весь пароль
//Ініціалізація використовуваних входів:
//Реле:
```

```
 pinMode(PIN_RELAY, OUTPUT);
 digitalWrite(PIN_RELAY, HIGH);
//Кнопка скидання пам'яті:
 pinMode(PIN_RESET,INPUT_PULLUP
 );
 key_reset.attach(PIN_RESET);
 key_reset.interval(5);
//Ініціалізація консолі послідовного виведення даних на
екран:
 Serial.begin(9600);
 while (! Serial);
 Serial.println(F("Start\r\n"));
//Ініціалізація основних модулів:
 lcd.init();
 lcd.backlight();
 SPI.begin();
 mfrc522.PCD_Init();
 pinMode(RED_LED, OUTPUT);
 pinMode(GREEN_LED, OUTPUT);
 digitalWrite(RED_LED, HIGH);
 digitalWrite(GREEN_LED, LOW);
 lcd.setCursor(4,0);
 lcd.print(L"ЧЕКАННЯ");
 lcd.setCursor(4,1);
 lcd.print(L"ДІЇ");
//Зчитуємо кількість ключів, значення має бути =>1 т.к
перший ключ //це майстер-ключ. У разі втрати, скинути
EEPROM і створити новий //мастер-ключ:
 if(keys_count > 0 and keys_count < 255) {
 keysRead();
 passRead();
 }
 else {
 keys_count = 0; //Визначення нового майстер-ключа
 Serial.println(F("The master key is not in memory. The
 first presentation to the
key will be the master!\r\n"));
 digitalWrite(PIN_RELAY, LOW);
 lcd.clear();
 lcd.setCursor(4,0);
 lcd.print(L"Вкажіть");
 lcd.setCursor(2,1);
 lcd.print(L"МАЙСТЕР-КЛЮЧ");
 }
}
```

```
// Функція циклу програми:
void loop()
{
 // Скидаємо сторожовий таймер
 мікроконтролера: wdt_reset();
 if(resetTimer > millis()+10000)
 resetTimer = 0; if(openTimer >
 millis()+10000) openTimer = 0;
 char key = keypad.getKey(); // задаємо функцію для
роботи з кнопками клавіатури
 if (key) // якщо натиснуто кнопку клавіатури
{
 Serial.println(key);
 squeaker(1, 2000, 100);
 presses=presses+1; //збільшуємо на одиницю рахунок
 кількості символів summ=summ+key;
 Serial.print("Pass"); Serial.println(summ);
//Виведення на дисплей знаків «*» при натисканні на
клавіатуру:
if(presses == 1 or shetch == 1)
{
lcd.clear();
lcd.setCursor(1,0);
lcd.print(" < PIN >");
lcd.setCursor(0,1);
lcd.print("*_");
}
if(presses == 2 or shetch == 2)
{
lcd.clear();
lcd.setCursor(1,0);
lcd.print(" < PIN >");
lcd.setCursor(0,1);
lcd.print("**_");
}
if(presses == 3 or shetch == 3)
{
lcd.clear();
lcd.setCursor(1,0);
lcd.print(" < PIN >");
lcd.setCursor(0,1);
lcd.print("***_");
}
if(presses == 4 or shetch == 4)
{
```

```
lcd.clear();
lcd.setCursor(1,0);
lcd.print(" < PIN >");
lcd.setCursor(0,1);
lcd.print("****");
}
 switch (dochange) //розгалуження на введення нового
 пароля (case 1) та перевірку (case
0)
 {
 case 0: //блок для нормальної роботи – перевірка
 правильності введення пароля
//У цьому випадку кнопки # і * призводять до скидання
кількості натискань:
 if (key=='#')
 {
 summ="";
 presses=0;
 Serial.println("# for RESET");
 squeaker(1, 500, 100);
 };
 if (key=='*')
 {
 summ="";
 presses=0;
 Serial.println("* for ENTER");
 squeaker(1, 500, 100);
 };
// Якщо правильний пароль і не було запиту на
 його зміну: if (summ==pass && change==0) {
 Serial.println("PASS
 OK"); summ="";
 presses=0;
 wrong=0;
 openTimer = millis()/1000;
 squeaker(2, 3500, 200, 100);
 digitalWrite(PIN_RELAY,
 LOW); allow();
 };
/ / Якщо натиснута комбінація для зміни пароля:
 if (summ=="0000") {
 Serial.println("Change pass go test");
 summ="";
 presses=0;
 wrong=0;
```

```
 change=1;
 squeaker(3, 700, 150);
 lcd.clear();
 lcd.setCursor(2,0);
 lcd.print(L"ЗМІНА ПАРОЛЯ");
 };
//Якщо правильний пароль і був запит на
 зміну коду if (summ==pass && change==1){
 Serial.println("Pass ok go change pass");
 summ="";
 presses=0;
 wrong=0;
 dochange=1;
 key = keypad.getKey();
 squeaker(4, 1000, 50);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(L"ЗМІНА ДОЗВОЛЕНА");
 };
//При зміні пароля - якщо введено повну кількість знаків
пароля, і він //помилковий, формується звуковий сигнал:
if (wrong==0 && presses==4 && change==1) {
 summ="";
 presses=0;
 wrong=wrong+1;
 Serial.println("Wrong_Pass");
 squeaker(4, 500, 50);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(L"НЕВІРНИЙ ПАРОЛЬ");
 };
//При зміні пароля - якщо двічі помилковий код, формуємо
звукові //сигнали
 if (wrong==1 && presses==4 && change==1) {
 summ="";
 presses=0;
 wrong=wrong+1;
 Serial.println("Attention!_2xWrong_Pass for change pas");
 squeaker(4, 500, 50);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(L"НЕВІРНИЙ ПАРОЛЬ");
 lcd.setCursor(3,1);
 lcd.print(L"ДРУГИЙ РАЗ");
 };
```

```
//При зміні пароля - якщо тричі помилковий код, виходимо
з режиму //зміни пароля:
 if (wrong==2 && presses==4 && change==1){
 summ="";
 presses=0;
 wrong=0;
 change=0;
 Serial.println("Attention!_3xWrong_Pass for change pas -
 cancel change pass");
 squeaker(1, 500, 1000);
 denied();
 };
/ / Якщо введено повну кількість символів пароля, але він
помилковий:
 if (presses==4 && change==0){
 summ="";
 presses=0;
 wrong=wrong+1;
 Serial.println("Wrong_Pass");
 squeaker(1, 500, 300);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(L"НЕВІРНИЙ ПАРОЛЬ");
 delay(1000);
 wait();
 };
//Якщо двічі помилковий пароль:
 if (wrong==2 && presses==0 && change==0) {
 Serial.println("Attention!_2xWrong_Pass");
 squeaker(1, 500, 500);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(L"НЕВІРНИЙ ПАРОЛЬ");
 lcd.setCursor(3,1);
 lcd.print(L"ДРУГИЙ РАЗ");
 delay(1000);
 wait();
 };
//Якщо три рази помилковий пароль:
 if (wrong==3 && presses==0 &&
 change==0) { summ="";
 presses=0;
 wrong=0;
```

```
Serial.println("Attention!_3xWr
ong_Pass"); squeaker(1, 500,
1000); denied();
};
break;
//Друга частина блоку зміни пароля, початок блоку зміни
пароля та його //записи в енергонезалежну пам'ять:
case 1:
if (key=='#') //якщо введено символ # скидаємо код
{
shetch = 1;
summ="";
Serial.println("# is not an option, reset");
squeaker(1, 500, 100);
}
else if (key=='*') //якщо введено символ * скидаємо код
{
shetch = 1;
summ="";
Serial.println("* is not an option, reset");
squeaker(1, 500, 100);
}
else if (shetch==1 && (key)) //Змінюємо перший символ
пароля
{
Serial.print("NewPass_symbol_one");
Serial.println(key); squeaker(1, 2000,
100);
shetch = 2; //збільшуємо на одиницю рахунок кількості
символів нового пароля EEPROM.write(1, key);
Serial.println(char(EEPROM.read(1))); //Промовляємо
перший знак пароля в порт
}
else if (shetch==2 && (key)) // Змінюємо другий символ
пароля
{
Serial.print("NewPass_symbol_two"); Serial.println(key);
EEPROM.write(2, key);
squeaker(1, 2000, 100);
shetch = 3; //збільшуємо на одиницю рахунок кількості
символів нового пароля
Serial.println(char(EEPROM.read(2))); // промовляємо
другий знак пароля в порт
}
```

```
else if (shetch==3 && (key)) // Змінюємо
третій символ пароля {
Serial.print("NewPass_symbol_three");
Serial.println(key);
EEPROM.write(3, key);
squeaker(1, 2000, 100);
shetch = 4; //збільшуємо на одиницю рахунок кількості
символів нового пароля
Serial.println(char(EEPROM.read(3))); // промовляємо
третій знак пароля в порт
}
else if (shetch==4 && (key)) // Змінюємо 4-
ий символ пароля {
Serial.print("NewPass_symbol_four"); Serial.println(key);
EEPROM.write(4, key);
squeaker(1, 2000, 100);
Serial.println(char(EEPROM.read(4))); // промовляємо
четвертий знак пароля в порт
String passnew = ""; // вводимо змінну, що містить
новий пароль.
passnew                          =
passnew+char(EEPROM.read(1));
passnew                          =
passnew+char(EEPROM.read(2));
passnew                          =
passnew+char(EEPROM.read(3));
passnew                          =
passnew+char(EEPROM.read(4));
passRead();
if (passnew==pass) // якщо новий пароль дорівнює старому
{
shetch = 1; // запитуємо інший новий пароль
summ="";
Serial.println("NewPass equal old pass, Reset");
squeaker(5, 600, 100);
lcd.clear();
lcd.setCursor(1,0);
lcd.print(L"СТАРИЙ ПАРОЛЬ");
}
else if (passnew=="0000") // якщо новий пароль дорівнює
комбінації для зміни пароля
{
shetch = 1; // Запитуємо інший
новий пароль summ="";
Serial.println("NewPass equal 0000, Reset");
```

```
 squeaker(5, 600, 100);
 lcd.clear();
 lcd.setCursor(1,0);
 lcd.print(L"СТАРИЙ ПАРОЛЬ");
 }
else {
//Привласнюємо паролю значення з енергонезалежної
пам'яті:
 pass="";
 pass = pass + char (EEPROM.read (1));
 pass = pass + char (EEPROM.read (2));
 pass = pass + char (EEPROM.read (3));
 pass = pass + char (EEPROM.read (4));
//Виведення в порт пароля для налагодження:
 Serial.println("Pass read test: ");
 Serial.println(char(EEPROM.read(1))); // промовляємо
 перший знак пароля в
порт
 Serial.println(char(EEPROM.read(2))); // промовляємо
другий знак пароля в порт
Serial.println(char(EEPROM.read(3))); // промовляємо
третій знак пароля в порт
Serial.println(char(EEPROM.read(4))); // промовляємо
четвертий знак пароля в порт
 Serial.print("Pass"); Serial.println(pass); //
 промовляємо пароль у порт
//Виходимо з циклу "case 1":
 dochange=0;
 change=0;
 presses=0;
 shetch = 1;
 summ="";
 squeaker(5, 900, 100);
 lcd.clear();
 lcd.setCursor(2,0);
 lcd.print(L"НОВИЙ ПАРОЛЬ");
 digitalWrite(GREEN_LED, HIGH);
 digitalWrite(RED_LED, LOW);
 delay(3000);
 digitalWrite(GREEN_LED, LOW);
 digitalWrite(RED_LED, HIGH);
 wait();
 break;
 }
 };
```

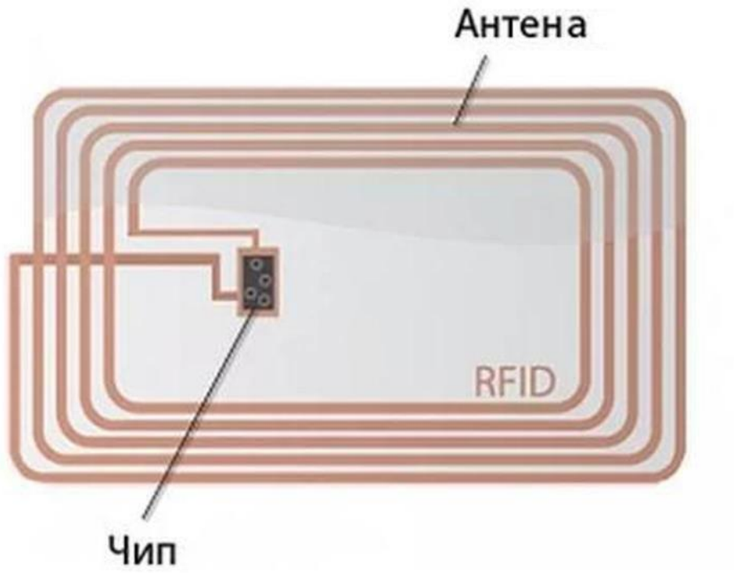
```
//кінець блоку для зміни пароля та його запису в
енергонезалежну пам'ять
}
}
// Очищення пам'яті:
key_reset.update();
if(key_reset.read() == HIGH) {
if(resetTimer == 0) resetTimer = millis();
else {
if((millis()-resetTimer)/1000 > 5) {
Serial.println(F("Launched memory cleaning"));
squeaker(4, 1600, 300, 200);
wdt_disable();
for(int i=5;
i<=EEPROM.length(); i++) {
EEPROM.write(i, 0);
if(!(i%50)) Serial.println(F("#")); else
Serial.print(F("#")); }
Serial.println(F("\r\nMemory cleaning is
completed\r\n"));
delay(1000);
resetFunc();
}
}
}
else if(resetTimer != 0) resetTimer = 0;
//Автоматичне закриття дверей через 5 секунд:
if(openTimer != 0) {
if(millis()/1000 - openTimer > 5) {
openTimer = 0;
digitalWrite(PIN_RELAY, HIGH);
digitalWrite(GREEN_LED, LOW);
digitalWrite(RED_LED, HIGH);
wait();
Serial.println("* closed lock\r\n");
}
}
//Якщо ключ відсутній чи читається, не виконуємо
подальший код:
if(!mfrc522.PICC_IsNewCardPresent()) {
//Очищення таймера входу в режим програмування, якщо
//зчитувач вільний:
if(modeTimer != 0) {
if(++modeClean > 5) modeTimer = modeClean = 0;
}
```

```
return;
}
if(!mfrc522.PICC_ReadCardSerial()) return;
//Зупиняємо режим очищення:
modeClean = 0;
//Створення майстер-ключа:
if(keys_count == 0) {
for(byte i=0; i<4; i++) EEPROM.write(i+5,
mfrc522.uid.uidByte[i]); EEPROM.write(0, keys_count
= 1); uidPrint(F("master key is created"));
digitalWrite(PIN_RELAY, HIGH);
keysRead();
squeaker(8, 1200, 100, 100);
delay(2000);
return;
}
//Перевірка ключа на відповідність:
bool access=false;
bool master = false;
for(byte i=0; i<keys_count; i++) {
for(byte b=0; b<4; b++) {
if(keyss[i][b] != mfrc522.uid.uidByte[b]) break;
if(b == 3) {
access = true;
if(i == 0) master = true;
//Зупиняємо перевірку i
= keys_count;
}
}
}
//Контроль доступу:
if(access and !mode and !master) {
//Доступ дозволено openTimer
= millis()/1000;
digitalWrite(PIN_RELAY,
LOW); squeaker(2, 3500, 200,
200); allow();
}
else if(!access and !mode and !master) {
//Доступ заборонено
squeaker(1, 500,
1000); denied();
}
//Режим програмування - запис ключа:
if(access and mode and !master) {
```

```
// Спроба запису існуючого ключа
uidPrint(F("error: key already exists
in eeprom")); squeaker(2, 500, 300);
lcd.clear();
lcd.setCursor(1,0);
lcd.print(L"КАРТА ВІДОМИЙ");
delay(2000);
wait();
}
else if(!access and mode and !master) {
//Записуємо новий ключ
// Максимум 255 ключів (з урахуванням
першого байта) if(keys_count < 255) {
for(byte i=0; i<4; i++) EEPROM.write(5 + keys_count*4 +
i,
mfrc522.uid.uidByte[i]);
EEPROM.write(0, ++keys_count);
uidPrint(F("add key in eeprom"));
keysRead();
lcd.clear();
lcd.setCursor(4,0);
lcd.print(L"ДОДАТОК");
lcd.setCursor(2,0);
lcd.print(L"НОВУ КАРТУ");
squeaker(2, 2200, 200, 200);
}
else // немає пам'яті для запису
{
uidPrint(F("error: not enough memory for recording
key!"));
squeaker(2, 500, 300);
}
delay(2000);
wait();
}
//Робота з майстер-ключом:
else if(access and master) {
//Майстер ключ у звичайному
режимі if(modeTimer == 0) {
modeTimer = millis()/1000;
if(!mode) {
openTimer = millis()/1000;
digitalWrite(PIN_RELAY, LOW);
```

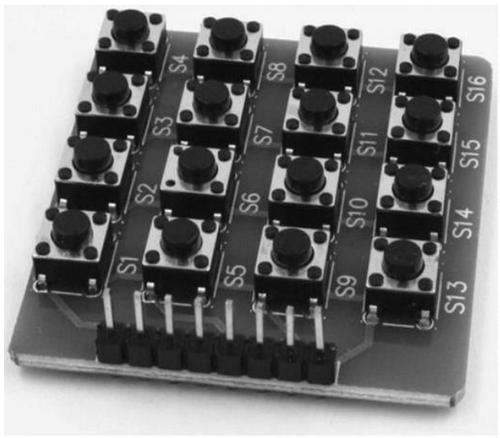
```
//Сигнал про наявність майстер ключа у
звичайному режимі uidPrint(F("MASTER
KEY"));
squeaker(2, 3200, 200,
200); allow();
}
}
else
{
if(millis()/1000 - modeTimer > modeProgTime and modeTimer
!= 0)
{
modeTimer = 0;
if((mode = !mode) == true)
{
//Вхід у режим програмування:
digitalWrite(PIN_RELAY, LOW);
uidPrint(F("MASTER PROGRAMMING MODE ON"));
squeaker(4, 1200, 200, 200);
}
else {
// Вихід із режиму програмування
digitalWrite(PIN_RELAY, HIGH);
uidPrint(F("MASTER PROGRAMMING MODE OFF"));
squeaker(4, 2200, 200, 200);
}
}
delay(5000);
wait();
}
//Майстер ключ утримується у зчитувача на 5 секунд
}
}
//Функція, що спрацьовує при правильному паролі/ключі:
void allow()
{
```

Елементи системи



RFID-мітка

Зчитувач RC522



Кнопкова та мембранна клавіатури 4x4

					13.02070849.51909 ПЛ1		
Зам.	Лист.	№ докум.	Підп.	Дата	Елементи системи Розробка багатофункціонального пристрою контролю доступу до приміщення	Лит.	Маса / Масшт.
Розроб.		Калістратов О.Г.					
Перев.		Ільяшенко М.Б.				Лист 1	Листів 1
Т.контр.							
Н.контр.		Щербак Н.В			НУ «Запорізька політехніка» КНТ-519		
Затв.		Кудерметов Р.К.					

Підключення модуля RC533 до Arduino Uno

MFRC522	Arduino Uno
RST	9
SDA	10
MOSI	11
MISO	12
SCK	13
3.3V	3.3V
GND	GND

Підключення RC522 до Arduino Uno

Ідентифікаційні номери RFID – міток

					13.02070849.51909 ПЛ2		
Зам.	Лист.	№ докум.	Підп.	Дата	Підключення модуля RC533 до Arduino Uno Розробка багатофункціонального пристрою контролю доступу до приміщення	Лит.	Маса / Масшт.
Розроб.		Калістратов О.Г.					
Перев.		Ільяшенко М.Б.				Лист 1	Листів 1
Т.контр.							
Н.контр.		Щербак Н.В			НУ «Запорізька політехніка» КНТ-519		
Затв.		Кудерметов Р.К.					

Основні етапи роботи пристрою

Відлагоджувальна інформація

```
UID: 60 121 172 213
MASTER PROGRAMMING MODE ON

UID: 101 229 204 101
add key in eeprom

KEYS COUNT: 2
------------------------------
KEY: 0 | 60 121 172 213
KEY: 1 | 101 229 204 101
------------------------------

UID: 60 121 172 213
MASTER PROGRAMMING MODE OFF
```

Режим програмування

Створення майстер-ключа

					13.02070849.51909 ПЛЗ		
Зам.	Лист.	№ докум.	Підп.	Дата	Основні етапи роботи пристрою Розробка багатофункціонального пристрою контролю доступу до приміщення	Лит.	Маса / Масшт.
Розроб.		Калістратов О.Г.					
Перев.		Ільяшенко М.Б.				Лист 1	Листів 1
Т.контр.							
Н.контр.		Щербак Н.В			НУ «Запорізька політехніка» КНТ-519		
Затв.		Кудерметов Р.К.					

Повна схема контролю доступу

					13.02070849.51909 ПЛ4		
Зам.	Лист.	№ докум.	Підп.	Дата	Повна схема контролю доступу Розробка багатофункціонального пристрою контролю доступу до приміщення	Лит.	Маса	Масшт.
Розроб.		Калістратов О.Г.						
Перев.		Ільяшенко М.Б.				Лист 1	Листів 1	
Т.контр.								
Н.контр.		Щербак Н.В				НУ «Запорізька політехніка» КНТ-519		
Затв.		Кудерметов Р.К.						