

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Інститут інженерії та інформаційних технологій
(повне найменування інституту, назва факультету)

Кафедра комп'ютерної інженерії та електромеханіки
(повна назва кафедри)

ДИПЛОМНА БАКАЛАВРСЬКА РОБОТА

на тему

ПРИСТРІЙ ВІДДАЛЕНОГО КЕРУВАННЯ ПРОПУСКНОЮ СИСТЕМОЮ

Виконав: студент групи БКІск-21

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Ярмоленко А.С.

(прізвище та ініціали)

Керівник д.т.н., доц. Стаценко В.В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Інститут інженерії та інформаційних технологій

Кафедра комп'ютерної інженерії та електромеханіки

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри КІЕМ

_____ проф. Злотенко Б.М.

“ _____ ” _____ 2023 року

ЗАВДАННЯ

НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Яроленко Ангеліні Сергіївні

(прізвище, ім'я, по батькові)

1. Тема дипломної бакалаврської роботи Пристрій віддаленого керування пропускнуою системою

Науковий керівник роботи Стаценко Володимир Володимирович,

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

Д.Т.Н., доцент

затверджені наказом вищого навчального закладу від 08.11.2022 № 224-уч.

2. Строк подання студентом роботи 1 червня 2023 року

3. Вихідні дані до дипломної бакалаврської роботи: програмне забезпечення для віддаленого керування пропускнуою системою; навчальна та методична література; державні стандарти.

4. Зміст дипломної бакалаврської роботи (перелік питань, які потрібно розробити): 1. Провести аналітичний огляд розроблених пропускнух систем. 2. Створити схему взаємодії компонентів та ПЗ для віддаленого керування пропускнуою системою. 3. Виконати розробку ПЗ для віддаленого керування пропускнуою системою та описати роботу програми.

5. Дата видачі завдання 01.02.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної бакалаврської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	10.02.2023	
2	Розділ 1. Аналітичний огляд розроблених пропускових систем	03.03.2023	
3	Розділ 2. Створення схеми взаємодії компонентів та програмного забезпечення для віддаленого керування пропусковою системою	07.04.2023	
4	Розділ 3. Розробка програмного забезпечення для віддаленого керування пропусковою системою та принцип роботи програми	12.05.2023	
6	Висновки	19.05.2023	
7	Оформлення дипломної бакалаврської роботи (чистовий варіант)	26.05.2023	
8	Здача дипломної бакалаврської роботи на кафедру для рецензування (за 14 днів до захисту)	29.05.2023	
9	Перевірка кваліфікаційної роботи на наявність текстових співпадінь та помилок (за 10 днів до захисту)	02.06.2023	
10	Подання дипломної бакалаврської роботи на затвердження завідувачу кафедри (за 7 днів до захисту)	06.06.2023	

Студент

_____ Ярошенко А. С.
(підпис) (прізвище та ініціали)

Науковий керівник роботи

_____ Стаценко В. В.
(підпис) (прізвище та ініціали)

Рецензент

_____ (підпис) _____ (прізвище та ініціали)

АНОТАЦІЯ

Ярмоленко А. С. Пристрій віддаленого керування пропускною системою. – Рукопис.

Дипломна бакалаврська робота за спеціальністю 123 Комп'ютерна інженерія, освітньою програмою «Комп'ютерні системи та мережі». – Київський національний університет технологій та дизайну, Київ, 2023 рік.

Дипломну бакалаврську роботу присвячено створенню програмного забезпечення для керування пропускною системою.

У роботі запропоновано пристрій віддаленого керування пропускною системою, розроблено схему взаємодії його компонентів та програмне забезпечення для віддаленого керування. Таке програмне забезпечення дозволить забезпечити роботу пропускної системи, а саме пропуск через турнікет за карткою з RFID-міткою. В роботі описано принцип роботи програмного забезпечення та надано інструкція з його використання. Окрім цього проведено дослідження вже створених пропускних систем та програмного забезпечення до них, описано процес створення схеми взаємодії компонентів та рекомендовані компоненти для неї.

Ключові слова: пропускна система, віддалене керування, Wi-Fi з'єднання, програмне забезпечення для віддаленого керування, RFID-мітка, Arduino.

ABSTRACT

Yarmolenko A. S. Device for remote control of the access system. – Manuscript.

Bachelor's thesis in the specialty 123 "Computer Engineering", educational program "Computer Systems and Networks". - Kyiv National University of Technology and Design, Kyiv, 2023.

The bachelor thesis is devoted to the creation of software for the control of the access system.

The paper proposes a scheme of interaction of the components of the access system with remote control and software for remote control. Such software will ensure the operation of the pass system, namely pass through the turnstile using a card with an RFID tag. The work describes the principle of operation of the software and instructions for its use. In addition, a study of already created access systems and software for them was conducted, the process of creating a component interaction scheme and recommended components for it were described.

Key words: access system, remote control, Wi-Fi connection, software for remote control, RFID tags, Arduino.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД РОЗРОБЛЕНИХ ПРОПУСКНИХ СИСТЕМ	9
1.1. Види пропускних систем.....	9
1.2. Призначення пропускних систем	10
1.3. Принципи роботи пропускних систем.....	11
Висновки до розділу 1	16
РОЗДІЛ 2. РОЗРОБКА СТРУКТУРИ ПРОПУСКНОЇ СИСТЕМИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО КЕРУВАННЯ ...	17
2.1. Створення загальної схеми	17
2.2. Вибір компонентів	22
2.3. Вимоги до створюваного програмного забезпечення.....	27
Висновки до розділу 2	32
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО КЕРУВАННЯ ПРОПУСКНОЮ СИСТЕМОЮ	Ошибка!
Закладка не определена.	
3.1. Передача даних між компонентами системи	33
3.2. Опис процесу роботи функцій програми для комп'ютера.....	42
3.3. Опис процесу роботи функцій програми для Arduino	50
Висновки до розділу 3	58
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТОК.....	66

ВСТУП

Актуальність роботи. Сучасний світ пропонує все більш розширені технології в усіх сферах діяльності. Кількість підприємств зростає із розвитком економіки, з'являється більше офісів та робочих приміщень. Загроз також стає більше, через що пропускні системи стають лише актуальнішими. Побудова пропускної системи та створення програми для керування нею у віддаленому режимі допоможе: забезпечити безпеку об'єктів від несанкціонованого доступу та крадіжок; контролювати та обліковувати робочий час персоналу, відвідувачів або клієнтів; оптимізувати роботу охоронних служб. Дротові системи втратили свою розповсюдженість, поступово замінюючись на бездротові, тому віддалене керування є дуже актуальним на часі. Завдяки з'єднанню через Wi-Fi підприємство заощаджує кошти на закупівлі елементів бездротової системи, отримує більш гнучку та мобільну систему, доступ до якої можна отримати з різних пристроїв.

Метою роботи є створення системи віддаленого керування пропускною системою.

Для досягнення поставленої мети у роботі було вирішено такі **задачі**:

- проведено аналітичний огляд розроблених пропускних систем;
- створено схему взаємодії компонентів та ПЗ для віддаленого керування пропускною системою;
- виконано розробку ПЗ для віддаленого керування пропускною системою;
- описано принцип роботи функцій, задіяних у програмі;
- описано роботу програми.

Об'єкт дослідження – віддалене керування пропускною системою.

Предмет дослідження – компоненти та програмне забезпечення для віддаленого керування пропускною системою.

Методи досліджень. Базою дослідження стали основні положення інформаційних систем та мереж, інформатики, електроніки та програмування.

Інформаційною базою досліджень є навчальна та методична література, державні стандарти, а також відкриті джерела Internet.

Практичне значення отриманих результатів. У роботі запропоновано створену схему взаємодії компонентів та програмного забезпечення для віддаленого керування пропускнуою системою та програмне забезпечення для віддаленого керування пропускнуою системою.

Апробація отриманих результатів. Результати роботи доповідались та отримали позитивну оцінку на II міжнародній науково-практичній інтернет конференції молодих учених та студентів «Електромеханічні, інформаційні системи та нанотехнології» (м. Київ, КНУТД, 20 квітня 2023 р.)

Структура та обсяг роботи. Дипломна робота бакалавра складається зі вступу, 3 розділів та висновків по них, загальних висновків, списку використаних джерел та додатків. Основний текст роботи викладений на 63 сторінках, містить 11 рисунків, 3 таблиці, список джерел з 54 найменувань. Загальний обсяг роботи, враховуючи додаток, складає 65 аркушів.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД РОЗРОБЛЕНИХ ПРОПУСКНИХ СИСТЕМ

1.1. Види пропускних систем

Система — множина взаємопов'язаних елементів, що утворюють єдине ціле, взаємодіють із середовищем та між собою, і мають мету [1].

Пропускні системи — це системи, що дозволяють контролювати доступ людей або транспортних засобів до певних територій, приміщень або об'єктів. Пропускні системи можуть використовувати різні методи ідентифікації, такі як пропуски, ключі, картки, браслети, біометричні дані тощо. Пропускні системи можуть бути автономними або мережевими, простими або складними, залежно від потреб та вимог користувача.[2]

Для вибору пропускної системи потрібно враховувати наступні фактори:

- Цілі та вимоги до системи, такі як рівень безпеки, кількість користувачів, точок доступу, ідентифікаторів тощо.
- Тип та метод ідентифікації, наприклад, пропуски, ключі, картки, браслети, біометричні дані тощо.
- Характеристики пристроїв для ідентифікації, контролю доступу та виконавчих пристроїв, наприклад, зчитувачі карток або браслетів, сканери відбитків пальців або обличчя, контролери доступу, двері, турнікети, шлагбауми тощо.
- Функціональність та зручність бази даних або програмного забезпечення для управління правами доступу та збору даних про події доступу.
- Сумісність та інтеграція пропускної системи з іншими системами, такими як CRM, ERP, банківські системи тощо.

- Бюджет та ресурси на побудову та обслуговування пропускної системи.[3]

1.2. Призначення пропускних систем

Пропускні системи потрібні для різних цілей, таких як:

- Забезпечення безпеки об'єктів від несанкціонованого доступу, крадіжок, тероризму та інших загроз.
- Контроль та облік робочого часу персоналу, відвідувачів або клієнтів.
- Оптимізація роботи охоронних служб, адміністративного персоналу та інших учасників процесу доступу.
- Покращення якості обслуговування та задоволення потреб користувачів.

Роль пропускних систем у безпеці полягає в тому, що вони допомагають запобігати або мінімізувати ризики, пов'язані з несанкціонованим доступом до об'єктів. Пропускні системи дозволяють:

- Визначати особистість та права доступу кожного, хто намагається увійти або вийти з об'єкта.
- Блокувати або обмежувати доступ тим, хто не має належних дозволів або не пройшов перевірку.
- Сповідати охоронні служби або інші відповідальні особи про спроби порушення режиму доступу або інші підозрілі події.
- Збирати та аналізувати дані про події доступу для покращення безпеки та контролю.[3]

Основні завдання системи контролю та управління доступом:

- обмеження несанкціонованого проникнення людей у робоче приміщення;
- контроль доступу персоналу до службових приміщень;
- контроль пересування співробітників підприємства упродовж робочого часу.[4]

1.3. Призначення пропускних систем

Пропускні системи працюють за наступним алгоритмом:

1. Користувач наближається до точки доступу, де встановлено пристрій для ідентифікації, наприклад, зчитувач карток або браслетів, сканер відбитків пальців або обличчя тощо.
2. Користувач пред'являє свій ідентифікатор, наприклад, пропуск, ключ, картку, браслет або свою біометричну характеристику.
3. Пристрій для ідентифікації зчитує інформацію з ідентифікатора та передає її до контролера доступу.
4. Контролер доступу перевіряє права доступу користувача за допомогою бази даних або програмного забезпечення.
5. Якщо користувач має право на доступ, контролер відкриває двері, турнікет, шлагбаум або інший виконавчий пристрій та дозволяє прохід. Якщо ні, контролер блокує доступ та виводить повідомлення про помилку.
6. Контролер реєструє подію доступу та зберігає її в базі даних або програмному забезпеченні для подальшого аналізу та звітності.

Для побудови пропускної системи потрібно виконати наступні кроки:

- Визначити цілі та вимоги до системи, такі як рівень безпеки, кількість користувачів, точок доступу, ідентифікаторів тощо.

- Обрати тип та метод ідентифікації, наприклад, пропуски, ключі, картки, браслети, біометричні дані тощо.
- Обрати пристрої для ідентифікації, контролю доступу та виконавчих пристроїв, наприклад, зчитувачі карток або браслетів, сканери відбитків пальців або обличчя, контролери доступу, двері, турнікети, шлагбауми тощо.
- Обрати базу даних або програмне забезпечення для управління правами доступу та збору даних про події доступу.
- Розробити проект пропускної системи з урахуванням архітектури об'єкта, маршрутів руху користувачів, мережевих з'єднань тощо.
- Встановити та налаштувати всі компоненти пропускної системи згідно з проектом.
- Провести тестування та налагодження пропускної системи для перевірки її функціональності та надійності.
- Забезпечити обслуговування та підтримку пропускної системи для її стабільної роботи.[3]

Компоненти систем контролю і управління доступу:

Ідентифікатори

Ідентифікатор - це унікальна ознака об'єкта, що дозволяє відрізнити його від інших об'єктів та звертатися до нього.[5] Ідентифікатор може бути ім'ям, номером, символом, кодом або іншою ознакою, яка цілком визначає сутність в наперед заданому просторі.[6]

Найбільшу популярність отримали ідентифікатори з безконтактною передачею даних RFID. А найпоширеніші форм-фактори - пластикова карта і брелок.

Типи ідентифікаторів:

- Безконтактні картки (EM-Marine або Mifare),

- Безконтактні брелоки (EM-Marine або Mifare),
- Безконтактні мітки (UHF).

В якості ідентифікатора може виступати не тільки карта або брелок, а й відбиток пальця, венозний малюнок (пальця, долоні), геометрія (форма) обличчя і т.д.

Зчитувачі

Зчитувач - невід'ємна складова СКУД. Даний електронний пристрій зчитує інформацію про об'єкт (наприклад, біометричні дані) і передає їх контролеру або терміналу для подальшого рішення про надання доступу. Існують різні моделі зчитувачів, які відрізняються за способом ідентифікації, по підтримуваним стандартам читання даних, за типом підключення.

Типи зчитувачів:

- Зчитувачі стандарту EM-Marine,
- Зчитувачі стандарту Mifare,
- Зчитувачі дальньої дії (UHF),
- Зчитувач відбитку пальців,
- Зчитувачі венозного малюнка пальця або долоні,
- Зчитувач з геометрії обличчя

Виконавчі пристрої

Забезпечують фізичну перешкоду, яка обмежує доступ до контрольованого об'єкту. Всі пристрої можна розділити на дві групи за принципом дії: електромеханічні і електромагнітні.

Типи виконавчих пристроїв:

- Електрозамки (електромагнітні та електромеханічні)
- Турнікети

Електрозамки мають різний клас захищеності і різні типи установки. Вони можуть використовуватися як всередині приміщення, так і зовні, можуть бути врізними і накладними.

Ключовою особливістю турнікетів є можливість пропускати на вході суто по одній людині. Даної можливості позбавлені всі інші засоби перегородження. Ця характеристика робить їх практично незамінними, коли стоїть мета ідентифікувати кожного, хто входить на об'єкт, що охороняється.[7]

Контролери

Контролери - це пристрої або системи, що здійснюють управління іншими пристроями або процесами за допомогою логічних, арифметичних або інших операцій.[8]

Ці пристрої вирішують наступні завдання:

- Управління виконуючими пристроями;
- Зберігання бази даних користувачів;
- Зберігання журналу подій.

Контролери бувають:

- автономними;
- мережевими.

Автономні контролери працюють виключно в автономному режимі. Це означає, що до них не можна підключити інші контролери, а якщо на об'єкті встановлено кілька таких пристроїв, то доведеться фізично підійти до кожного, перевести його в режим програмування і додати або видалити карти. Основною привабливою особливістю автономних контролерів є їх доступна ціна.

Мережеві контролери або термінали, скільки б їх не було, з легкістю об'єднуються в одну мережу, і, як правило, підключаються до комп'ютера за встановленим програмним забезпеченням, з якого легко і зручно можна керувати і робити онлайн-моніторинг всієї системи доступу.

Кнопки виходу

Будь-яка система контролю доступу не може обійтися без простого, але такого важливого елемента, як кнопка виходу. Кнопки виходу використовуються для відкриття дверей, хвіртки, шлагбаума, воріт і т. д.

Тип кнопок виходу:

- Нормально відкриті (NO) - даний тип кнопок працює на замикання і часто використовується з електромеханічними замками, засувками, контролерами.
- Нормально закриті (NC) - такі кнопки працюють на розмикання і використовуються при роботі з магнітними замками.
- Універсальні (NC / NO) - це найбільш популярний тип кнопок, так як даний вид може працювати на замикання і розмикання.

Важливим фактором є спосіб натискання на кнопку. Найбільш звичним і поширеним є механічний спосіб натискання. У випадку з сенсорною кнопкою необхідно тільки злегка доторкнутися до провідної поверхні, щоб змінився опір контролера і сталося відкриття дверей. Безконтактна кнопка виходу має інфрачервоний датчик, що реагує на наближення руки. Плюсом цього рішення є гігієнічність: безконтактної кнопки з інфрачервоним сенсором не потрібно торкатися, досить просто змахнути перед пристроєм рукою - і двері відкриваються.

Дверні доводчики

Дверні доводчики необхідні в системах контролю доступу, щоб після людини, яка проходить, випадково не залишилися відчинені двері.[7]

Принцип роботи системи пропуску за картками:

Співробітник офісу або виробничого підприємства при вході на об'єкт пред'являє (прикладає) ідентифікатор (карту, ключ, брелок) до зчитувача. Зчитувач – електронне обладнання, призначене для зчитування інформації з карт або електронних ключів. Код повинен збігатися з даними, закладеними в контролері, тим самим відбувається розпізнавання ключа і надання допуску.

Програма контролю доступу – це централізоване управління контролерами СКУД за допомогою персонального комп'ютера.[9]

Висновки до розділу 1

1. Проаналізовано види, структуру та принцип роботи пропускних систем.
2. Визначено компоненти, що входять до складу пропускних систем та їх характеристики.
3. Проаналізовано підходи до розробки пропускних систем;
4. Визначено коло задач, що виконують пропускні системи, їх роль у безпеці підприємства, переваги та недоліки.
5. Встановлено, що бездротовий тип підключення найбільш вигідний, саме тому його обрано для системи, що представлена у роботі.
6. Система з RFID картками є простою у виконанні та відповідає поставленим у роботі вимогам.

РОЗДІЛ 2. РОЗРОБКА СТРУКТУРИ ПРОПУСКНОЇ СИСТЕМИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО КЕРУВАННЯ

2.1. Створення загальної схеми

Контроль доступу - це функція відкритої системи, що забезпечує технологію безпеки, яка дозволяє або забороняє доступ до певних типів даних, засновану на ідентифікації суб'єкта, якому потрібен доступ, і об'єкта даних, що є метою доступу. Основними механізмами контролю доступу є ідентифікація і аутентифікація.[10]

Автоматизована система контролю доступу - це електронна або електронно-механічна система, що призначена для надання дозволу на прохід персоналу, проїзд транспортних засобів або переміщення матеріалів через вхід (вихід) контрольно-пропускних пунктів зон обмеження доступу. Така система може бути автономною, центральною (мережевою) або універсальною.[11]

Компоненти в системі контролю доступу можуть включати: контролер - пристрій, що керує виконавчими пристроями і обмінюється інформацією з центральним пунктом охорони; зчитувач - пристрій, що зчитує інформацію з носія (картки, брелока, біометричного сенсора тощо) і передає її контролеру; замок - пристрій, що блокує або розблоковує прохід (магнітний або електромеханічний); кнопка виходу - пристрій, що дозволяє вийти з контрольованої зони без носія; блок живлення - пристрій, що забезпечує стабільне живлення для всього обладнання; системний комп'ютер - пристрій, що програмує контролери СКД на дозвіл чи заборону проходів і створює автоматизоване робоче місце адміністратора СКД.[12]

Турнікети - це пристрої, які обмежують доступ людей до певної території або приміщення. Вони можуть бути різних типів: триподи, хвіртки, повноростові, розсувні тощо. Турнікети можуть працювати в ручному або автоматичному режимі, інтегруватися з системами контролю доступу (СКУД) і використовувати різні методи ідентифікації: картки, браслети, відбитки пальців тощо.[15]

Пристрій керування на базі мікроконтролера - це пристрій керування, який використовує мікроконтролер як основний елемент логіки і обчислень. Мікроконтролер - це мікросхема, що включає мікропроцесор, оперативну та постійну пам'ять, порти вводу-виводу і блоки зі спеціальними функціями[16]. Пристрої керування на базі мікроконтролера можуть бути реалізовані апаратно або мікропрограмно[17].

Апаратне керування забезпечується логічними схемами, що формують керуючі сигнали в залежності від коду поточної команди і стану пристрою. Апаратне керування має високу швидкодію, але ускладнює конструкцію пристрою.

Мікропрограмне керування забезпечується мікропрограмами, що зберігаються в постійній пам'яті мікроконтролера і визначають послідовність керуючих сигналів для кожної команди. Мікропрограмне керування спрощує конструкцію пристрою, але знижує швидкодію.

Пристрої керування на базі мікроконтролера застосовуються для реалізації різних завдань, таких як керування електронними пристроями, системами контролю і автоматизації, вимірювальними приладами тощо[18].

RFID-модулі - це пристрої, які дозволяють зчитувати і записувати дані з RFID-міток за допомогою радіочастотного випромінювання. RFID-мітка - це маленький пристрій, який містить мікрочип і антену і зберігає унікальний ідентифікатор або іншу інформацію про об'єкт, до якого вона прикріплена[19]. RFID-модулі можуть бути підключені до різних систем

керування, контролю або автоматизації для виконання різних завдань, таких як:

- Ідентифікація та відстеження товарів, тварин, людей тощо;
- Автоматична оплата за проїзд, паркування, доступ до територій тощо;
- Збереження та передача даних про стан об'єктів, їх параметри, історію тощо[20];
- Створення інтерактивних пристроїв, іграшок, гаджетів тощо[21].

RFID-модулі можуть мати різну конструкцію, розмір, потужність, дальність дії, частоту роботи тощо в залежності від призначення та умов експлуатації[22].

RFID-технологія - це технологія автоматичної ідентифікації, яка використовує радіохвилі для збору і передачі даних. RFID-технологія працює за таким принципом:

RFID-мітка, яка прикріплена до об'єкта, випромінює радіосигнал з унікальним ідентифікатором або іншою інформацією про об'єкт;

RFID-зчитувач, який знаходиться у зоні дії мітки, отримує радіосигнал і перетворює його в цифровий формат;

RFID-система, яка підключена до зчитувача, зберігає, обробляє і використовує отриману інформацію для різних цілей.

RFID-технологія має багато переваг перед іншими методами ідентифікації, такими як штрих-коди або магнітні смуги. Деякі з них:

- RFID-мітки не потребують прямої видимості або контакту з зчитувачем.
- RFID-мітки можуть мати багато режимів роботи: тільки читання, читання/запис, пасивні, активні тощо.

- RFID-мітки можуть містити багато даних і мати розширену функціональність: наприклад, змінювати дані в залежності від умов або виконувати додаткові операції.
- RFID-зчитувач може одночасно зчитувати кілька міток, що збільшує швидкодію і точність ідентифікації[23].

Arduino - це ефективний засіб розробки програмованих електронних пристроїв, які, на відміну від персональних комп'ютерів, орієнтовані на тісну взаємодію з навколишнім світом. Arduino - це відкрита програмована апаратна платформа для роботи з різними фізичними об'єктами і є простою платою з мікроконтролером, а також спеціальним середовищем розробки для написання програмного забезпечення мікроконтролера[24].

Пост охорони на підприємстві - це спеціально обладнане приміщення або споруда, де розміщуються працівники охоронної фірми, які забезпечують контроль за в'їздом та виїздом людей і транспорту на територію підприємства, а також за дотриманням правил внутрішнього розпорядку і безпеки[25].

Сервер - це апаратне або програмне забезпечення, що надає певні послуги іншим пристроям або програмам - клієнтам. Сервер може бути:

- Виділеним комп'ютером, що працює автономно і забезпечує доступ до своїх ресурсів і сервісів через локальну або глобальну мережу. Наприклад, файловий сервер, веб-сервер, поштовий сервер тощо.
- Програмою, що виконується на комп'ютері і обробляє запити від клієнтських програм за допомогою певного протоколу. Наприклад, сервер баз даних, сервер друку, сервер ігор тощо.

Сервери мають різні характеристики та спеціалізації, залежно від їх призначення, функціональності, надійності, продуктивності та

масштабованості. Сервери також можуть бути розміщені та обслуговувані в різних місцях та способами[26].

Сервер бази даних - це програма або комп'ютер, що зберігає, обробляє і надає доступ до даних, організованих у вигляді бази даних.

База даних - це сукупність даних, що мають певну структуру і логічний зв'язок між собою.

Система управління базами даних (СУБД) - це програмне забезпечення, що дозволяє створювати, модифікувати, запитувати і адмініструвати бази даних.

База даних — це певний набір даних, які пов'язані між собою спільною ознакою або властивістю, та впорядковані, наприклад, за алфавітом.

Щоб створити запит до бази даних часто використовують Structured Query Language. SQL дає змогу додавати, редагувати та видаляти інформацію, що міститься у таблицях[30].

В роботі передбачено створення СКД, структурна схема якої показана на рис. 1. До її складу входять: турнікети, що забезпечують фізичне обмеження доступу; пристрій керування на базі мікроконтролера (МК) AtMega 326, що входить до складу платформи Arduino[13], комп'ютер посту охорони та сервер бази даних.

Ключем, що дозволяє відвідувачам пройти через турнікет, є картка з RFID-міткою.

Коли мітку підносять до турнікету, пристрій керування зчитує інформацію з неї за допомогою RFID-модуля RC522, основою якого є мікросхема MFRC522. Модуль працює на частоті 13,56 МГц, а обмін даними з мікроконтролером здійснюється через SPI-інтерфейс. Далі МК передає інформацію з RFID-мітки на комп'ютер центрального посту охорони через Wi-Fi мережу.[14] Для цього до складу пристрою керування входить модуль ESP8266. Це дозволяє підключити до одного комп'ютера одночасно декілька турнікетів. Програма, що встановлена на комп'ютері посту охорони, шукає

дані RFID-мітки у базі даних та у випадку успішного проходження перевірки надсилає до МК команду розблокування відповідного турнікету.

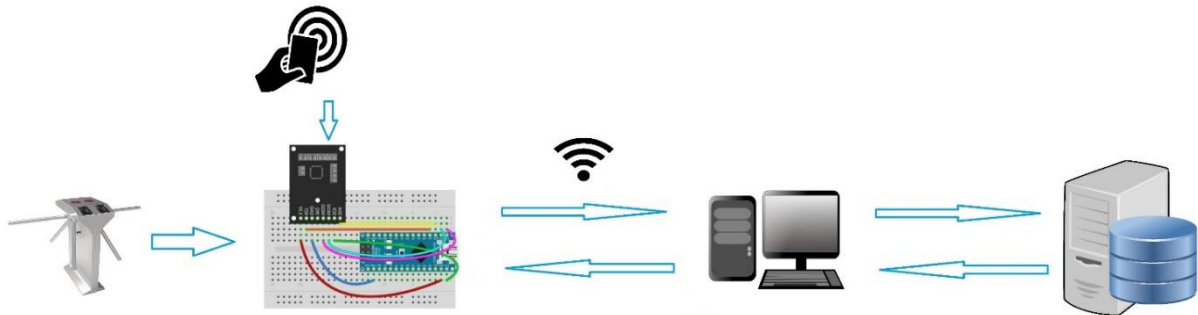


Рис. 2.1. Структурна схема системи контролю доступу

Запропонована система дозволяє реалізувати наступні функції:

- формування індивідуальних і групових повноважень доступу;
- облік робочого часу співробітників (перебування їх у певному приміщенні);
- облік доступу сторонніх осіб із зазначенням останнього місця їх перебування;
- контроль стану всіх приміщень (відображення плану об'єкта з коментарями в протоколі подій);
- можливість окремого блокування та розблокування будь-якого з терміналів.

2.2. Вибір компонентів

RFID-модулі - це пристрої, які дозволяють зчитувати і записувати інформацію на безконтактні картки або мітки за допомогою радіохвиль. RFID-модулі використовуються для ідентифікації осіб, тварин, товарів, автомобілів та інших об'єктів. RFID-модулі можуть мати різну форму, розмір, частоту, дальність дії та функціональність[23].

RFID-модуль RC522 - це пристрій, який дозволяє зчитувати і записувати інформацію на безконтактні картки або мітки за допомогою

радіохвиль частотою 13,56 МГц. RFID-модуль RC522 підтримує стандарт ISO 14443A і може працювати з різними типами карт, такими як Mifare1 s50, Mifare1 s70 , MIFARE Ultralight, Mifare Pro, MIFARE DESFire. RFID-модуль RC522 підключається до мікроконтролерів за допомогою інтерфейсу SPI, а також може використовувати інтерфейси I2C та UART. RFID-модуль RC522 має вивід преривання IRQ, який дозволяє опитувати модуль тільки тоді, коли приклали картку[31].



Рис. 2.1 - RFID-модуль RC522

Основні технічні характеристики RFID-модуля RC522 такі[32]:

- Основна мікросхема: RC522;
- Напруга живлення: 3,3 В;
- Споживаний струм: 13 – 26 мА;
- Дальність зчитування: 0 ~ 60 мм;
- Інтерфейс: SPI, максимальна швидкість передачі 10 Мбіт/с;
- Робоча температура: від -20 до 80 °С;
- Розмір модуля: 40 x 60 мм;
- Вага комплекту: 22 г.

Для того, щоб працювати з RFID-модулем RC522, потрібно підключити модуль до плати Arduino або іншого мікроконтролера за допомогою проводів DuPont. Схема підключення наведена в таблиці 2.1[32]:

Таблиця 2.1

Вивод модуля	Вивод Arduino
VCC	3.3V
RST	D9
GND	GND
IRQ	Не підключено
MISO	D12
MOSI	D11
SCK	D13
SDA	D10

Мікросхема - це електронний пристрій (схема) довільної складності, зібрана на напівпровідниковій платформі або плівці[33]. Мікросхема містить пасивні і активні радіоелектронні компоненти, такі як транзистори, резистори, конденсатори, діоди тощо[34]. Мікросхема може виконувати різні функції, такі як логічні операції, обробка сигналів, пам'ять, генерація частоти і т.д. Мікросхеми використовуються в багатьох галузях електроніки, таких як комп'ютери, мобільні телефони, побутова техніка, медичне обладнання і т.д.

Можна виділити два основних типи мікросхем: аналогові і цифрові. Аналогова мікросхема працює з безперервним сигналом і може змінювати його форму, амплітуду або частоту. Цифрова мікросхема працює з дискретним сигналом і може виконувати логічні операції над ним[36]. Існують також змішані мікросхеми, які поєднують аналогові і цифрові функції[37].

Мікросхема MFRC522 - це чіп для безконтактного зчитування і запису інформації на RFID-картки або мітки стандарту ISO/IEC 14443 A/MIFARE і NTAG. Мікросхема MFRC522 має вбудований передавач, який може приводити в дію антену для зв'язку з RFID-картками і транспондерами без додаткових активних компонентів. Мікросхема MFRC522 також має приймач, який забезпечує надійне і ефективне демодулювання і декодування сигналів від RFID-карток і транспондерів. Мікросхема MFRC522 підтримує шифрування продуктів MF1xxS20, MF1xxS70 і MF1xxS50 у режимі читання/запису. Мікросхема MFRC522 підтримує безконтактну комунікацію і використовує вищі швидкості передачі MIFARE до 848 кбіт/с в обох напрямках.

Мікросхема MFRC522 має наступні хост-інтерфейси:

- Серійний периферійний інтерфейс (SPI)
- Серійний UART (схожий на RS232 з рівнями напруги, залежними від напруги живлення на виводах)
- Інтерфейс I2C [38]

Можна підключити мікросхему MFRC522 до плати Arduino або іншого мікроконтролера за допомогою проводів DuPont. Схема підключення наведена в таблиці 2.2[39]:

Таблиця 2.2

Вивод модуля	Вивод Arduino
VCC	3.3V
RST	D9
GND	GND
IRQ	Не підключено
MISO	D12
MOSI	D11
SCK	D13
SDA	D10

Модуль ESP8266 - це мікроконтролер з WiFi інтерфейсом, який може використовуватися як WiFi модуль для Arduino або інших мікроконтролерів, або як самостійний пристрій, який виконує код програм з вбудованої флеш-пам'яті. Модуль ESP8266 підтримує стандарт IEEE 802.11 b/g/n, може працювати в режимах точки доступу або станції, має повний стек TCP/IP і може підключатися до локальної мережі або до Інтернету.

Мікроконтролери - це спеціалізовані мікроелектронні програмовані пристрої, що включають мікропроцесор, пам'ять, порти вводу-виводу і блоки зі спеціальними функціями. Мікроконтролери використовуються для керування різними електронними пристроями, такими як телефони, пральні машини, іграшки, верстати, домашня техніка, системи автоматизації тощо. Мікроконтролери можуть виконувати прості або складні завдання залежно від їх архітектури, системи команд, об'єму пам'яті і периферії. Мікроконтролери можуть працювати самостійно або співпрацювати з іншими мікроконтролерами або мікропроцесорами[41].

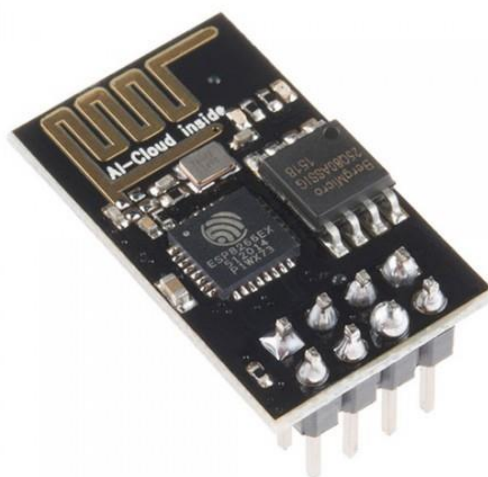


Рис. 2.2 - Модуль ESP8266

Модуль ESP8266 можна підключити до плати Arduino або іншого мікроконтролера за допомогою проводів DuPont. Схема підключення наведена в таблиці 2.3:

Таблиця 2.3

Вивод модуля	Вивод Arduino
VCC	3.3V
CH_PD	3.3V
GND	GND
Tx	Rx
Rx	Tx

Модуль ESP8266 живиться від 3.3 В і має 32-бітний процесор Tensilica L106 з тактовою частотою 80-160 МГц, 32 КБ оперативної пам'яті для інструкцій та 80 КБ для даних, 17 портів вводу-виводу загального призначення (GPIO), один 10-бітний перетворювач АЦП для сигналів, спеціальні висновки UART, а також інтерфейси SPI та I2C. Модуль ESP8266 також має зовнішню флеш-пам'ять QSPI від 512 КБ до 16 МБ залежно від моделі модуля[35].

2.3. Вимоги до створюваного програмного забезпечення

Програмне забезпечення - це сукупність програм, що використовуються для роботи з комп'ютерами та пов'язаними з ними пристроями. Програма - це впорядкований набір команд, які розповідають комп'ютеру, як працювати. Програмне забезпечення може бути написане на різних мовах програмування, які перекладаються на машинну мову за допомогою компілятора або інтерпретатора[41]. Програмне забезпечення може бути також називатися програмним продуктом, якщо воно розроблене для вирішення задачі масового попиту та призначене для постачання користувачам[42].

Програмне забезпечення можна класифікувати за різними критеріями, наприклад:

- За призначенням: системне (забезпечує роботу комп'ютера та інших програм), інструментальне (надає засоби для розробки та тестування інших програм), прикладне (вирішує конкретні задачі користувачів).
- За способом поширення: комерційне (платне), безкоштовне (безкоштовно доступне для всіх), вільне (з відкритим кодом і правом на модифікацію) .
- За способом використання: локальне (встановлюється на обладнанні користувача), веб-орієнтоване (доступне через інтернет), хмарне (зберігається і обробляється на віддалених серверах)[41].

Функції програмного забезпечення для віддаленого керування пропускною системою можуть бути різними в залежності від конкретного продукту та потреб користувача. Однак, загальною метою такого програмного забезпечення є надання можливості віддалено керувати процесом пропуску людей та транспорту на об'єкті або території за допомогою комп'ютера або мобільного пристрою. Для цього таке програмне забезпечення повинно мати такі функції:

- Встановлення безпечного з'єднання з віддаленим комп'ютером або пристроєм, який керує пропускною системою.
- Відображення стану пропускної системи, такого як кількість людей та транспорту, що пройшли або очікують на пропуск, час очікування, статистика за період тощо.
- Керування параметрами пропускної системи, такими як режим роботи, правила пропуску, методи ідентифікації, база даних користувачів тощо.

- Формування звітів та сповіщень про події, що виникають у пропускній системі, таких як порушення правил, спроби несанкціонованого доступу, аварійні ситуації тощо.
- Передача файлів та голосове спілкування з персоналом, який обслуговує пропускну систему.

Переваги використання програмного забезпечення для віддаленого керування пропускнуою системою можуть бути такими:

- Збільшення продуктивності та ефективності роботи персоналу, який обслуговує пропускну систему, оскільки вони можуть отримувати інструкції та підтримку віддалено.
- Зменшення витрат на паливо, час та ресурси, які необхідні для фізичного переїзду до місця розташування пропускнуої системи.
- Поліпшення якості обслуговування та задоволеності клієнтів, які користуються пропускнуою системою, оскільки вони можуть отримувати швидку та професійну допомогу в разі потреби.
- Забезпечення безпеки та контролю над пропускнуою системою, оскільки можна виявляти та запобігати спробам несанкціонованого доступу, порушень правил, аварійних ситуацій тощо.
- Збереження історичних даних та статистики про роботу пропускнуої системи, які можна аналізувати та використовувати для покращення її функціонування.

Для реалізації поставлених в роботі задач було обрано мову програмування C++ та серверно-клієнтську систему. Інструментом для створення програми обрано програмне забезпечення C++ Builder 11.

Створення додатків здійснюється за допомогою програмного забезпечення, **Інтегроване середовище розробки (IDE)**. Програміст використовує інтегроване середовище з усіма важливими ресурсами,

наданими IDE. Він покладається на графічний інтерфейс для розробки програмних додатків.

Творці швидко розвиваються, не обов'язково переходячи на іншу програму. Однак це відбувається за допомогою певних популярних практик, таких як налагодження, контроль версій і перегляд структури даних. Це призводить до підвищення ефективності за рахунок скорочення періоду вивчення мови та надання ідентичних інтерфейсів користувача (UI).

Компоненти інтегрованого середовища розробки (IDE)

Протягом тривалого часу використовуються такі програми, як IDE, і перейшли від програмного забезпечення для тестування та налагодження до інтегрованої форми програми, яка може діяти як розширення для розробника.

Текстовий редактор

У більшість IDE додано текстовий редактор, який використовується для керування вихідними кодами та записом.

Більшість програм мають базовий інтерфейс, який точно визначає певний синтаксис, однак інші містять візуальні компоненти, які дозволяють користувачам перетягувати функції інтерфейсу.

Інструменти для завершення коду

Виявляючи та включаючи звичайні функції коду стратегічно й розумно, інструменти завершення коду допомагають розробникам. Загалом, це економить час програмістів під час програмування та зменшує кількість помилок та бар'єрів.

Компілятори

Мови програмування, такі як двійковий код, перетворюються в машиночитані формати за допомогою компіляторів. Машинний код спочатку аналізується, щоб переконатися, що він правильний, а потім компілятор підвищує продуктивність шляхом аналізу та оновлення коду.

Підтримка мови програмування

Хоча деякі IDE дозволяють використовувати багато мов, деякі прихильні до одного програмування. Тому програмісти звужують свої параметри IDE на основі мови, на якій вони будуть програмувати. Прикладами є інструменти IDE для Ruby, Python і Java.

Відладчик

Програмісти знаходять і виправляють помилки за допомогою інструментів налагодження. Вони копіюють реальні сценарії для аналізу продуктивності та ефективності. Загалом, програмісти та інженери з програмного забезпечення проводять всебічний тест частин коду, щоб знайти помилки та помилки перед випуском програми.

Інтеграції та плагіни

З огляду на назву «інтегроване середовище розробки», не дивно, що інтеграції слід враховувати при оцінці IDE. Інтеграція інших інструментів для побудови з IDE підвищить процеси побудови та ефективність, оскільки IDE діє як портал розробки, таким чином запобігаючи завадам і перешкодам, які можуть бути викликані поганою інтеграцією.

Різні форми інтегрованого середовища розробки (IDE)

Інтегроване середовище розробки (IDE) міститься у всіх мовах програмування, і кожна з них має унікальні функції, які роблять програмування високоякісного коду простішим, швидким та ефективним.

Багатомовні IDE – це IDE, які підтримують різні мови програмування. Початківці розробники можуть мріяти про знання того, як використовувати багатомовну IDE (з наміром покращити свої навички пізніше). Візуальна студія, яка є гарним прикладом багатомовної IDE, яку можна використовувати, популярна завдяки своїм широким можливостям і стійкій підтримці розширень і оновлень. Процес включення розширення не складний, ніж запровадження підтримки нової мови програмування[45].

Visual Studio 2022 - це найповніша інтегрована середовище розробки для .NET і C++ розробників на Windows для створення веб-, хмарних, настільних, мобільних додатків, сервісів та ігор. Ви можете завантажити безкоштовну версію Community або скористатися безкоштовною пробною версією Professional або Enterprise. Ви також можете отримати доступ до останніх функцій, яких ще немає в основному релізі, завантаживши Preview. Visual Studio 2022 має багато переваг, таких як 64-бітна IDE, AI-підсилення коду, спільне кодування в режимі реального часу, інтегрована підтримка Git та GitHub та багато іншого[49].

Висновки до розділу 2

1. Розроблено структуру СКД, показано принцип взаємодії її компонентів.
2. В якості складових елементів СКД обрано мікросхему MFRC522, RFID-модуль RC522, Wi-Fi модуль ESP8266, плату Arduino Nano, наведено їх основні технічні характеристики.
3. Обрано інструменти розробки програмного забезпечення, а саме: Arduino IDE та Visual Studio 2022.
4. Визначено основні функції програми віддаленого керування пропускнуою системою.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО КЕРУВАННЯ ПРОПУСКНОЮ СИСТЕМОЮ

3.1. Передача даних між компонентами системи

Wi-Fi - це бездротова технологія мережі, яку використовують пристрої для з'єднання з інтернетом або іншими пристроями. Вона використовує радіохвилі для передачі даних безпроводним шляхом і підтримується різними сучасними електронними пристроями, включаючи комп'ютери та смартфони. Wi-Fi стала однією з найпопулярніших технологій.

Основною вимогою для Wi-Fi є пристрій, який приймає і передає бездротовий сигнал, зазвичай маршрутизатор, але іноді телефон або комп'ютер. У типовому будинку або малому бізнесі маршрутизатор приймає і потім передає підключення до інтернету, яке надходить від постачальника послуг інтернету за межами мережі. Він надає цей сервіс найближчим пристроям, які можуть отримати бездротовий сигнал. Багато будинків мають один маршрутизатор і кілька пристроїв, включаючи смартфони, настільні та портативні комп'ютери, планшети та розумні телевізори, які отримують своє підключення до інтернету через Wi-Fi від цього маршрутизатора. У деяких випадках користувач без маршрутизатора може налаштувати телефон або комп'ютер як точку доступу Wi-Fi, щоб поділитися бездротовим або проводним підключенням до інтернету з іншими пристроями, подібно до того, як працює маршрутизатор. Незалежно від того, як використовується Wi-Fi або джерела підключення, результат завжди один і той же: бездротовий сигнал, який дозволяє іншим пристроям підключатися до основного передавача для комунікації, розваг, передачі файлів, голосових повідомлень або іншої передачі даних[50].

Для забезпечення зв'язку через Wi-Fi встановлено ESP8266.

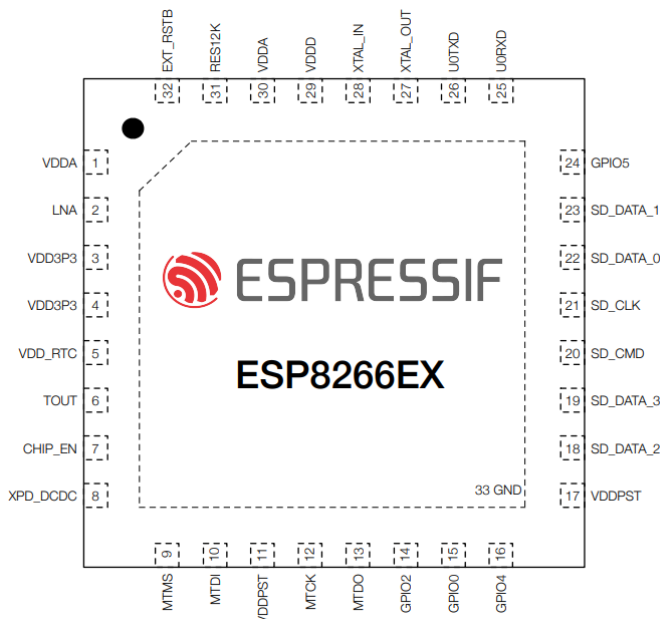


Рис. 3. 1 - Розміщення виводів ESP8266

RFID мітка — це мініатюрний пристрій, що запам'ятовує, який складається з інтегральної схеми (мікрочипу) для зберігання та обробки інформації та антени. У пам'яті мікрочипу зберігається інформація, а антена передає та отримує сигнал. Антенна зчитувача випромінює електромагнітні хвилі, внаслідок цього здійснюється живлення мітки. У результаті чого мітка активується та передає інформацію пристрою, що зчитує. Немає потреби у контакті або прямій видимості між зчитувачем та міткою, тому що радіосигнал легко проникає через більшість матеріалів. Таким чином, мітки можуть бути сховані усередині тих об'єктів, що підлягають ідентифікації. У пам'яті RFID міток міститься їх унікальний ідентифікаційний код. Деякі мітки мають перезаписувану пам'ять. Пристрій, що зчитує (зчитувач або рідер) — електронні пристрої, які зчитують інформацію з міток і записують у них дані. Ці пристрої можуть бути як постійно підключеними до облікової системи, так і працювати автономно. Облікова система — програмне забезпечення, яке накопичує та аналізує отриману з міток інформацію та пов'язує всі елементи в єдину систему [47].

Для того, щоб система могла зчитувати інформацію з мітки, в турнікет на мікросхему Arduino додано RFID-модуль RC522.

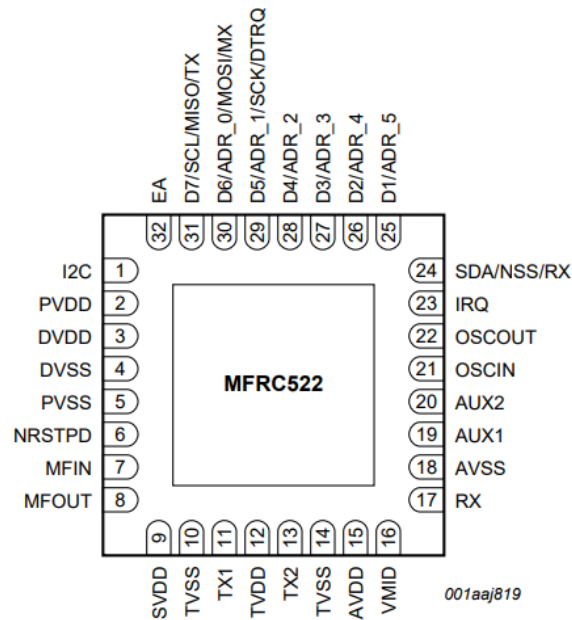


Рис. 3.2 - Розміщення виводів MFRC522

Для початку роботи з модулем, необхідно підключити його до Arduino, додати прошивку. Схема підключення представлена на рис. 3.1. Далі - підключити до Wi-Fi мережі та налаштувати передачу даних між комп'ютером посту охорони (сервером) та турнікетом (клієнтом).

Після підключення зібраної схеми до комп'ютера, потрібно перемкнути модуль в режим програмування. Для здійснення прошивки необхідно завантажити на налаштувати програму Arduino IDE.

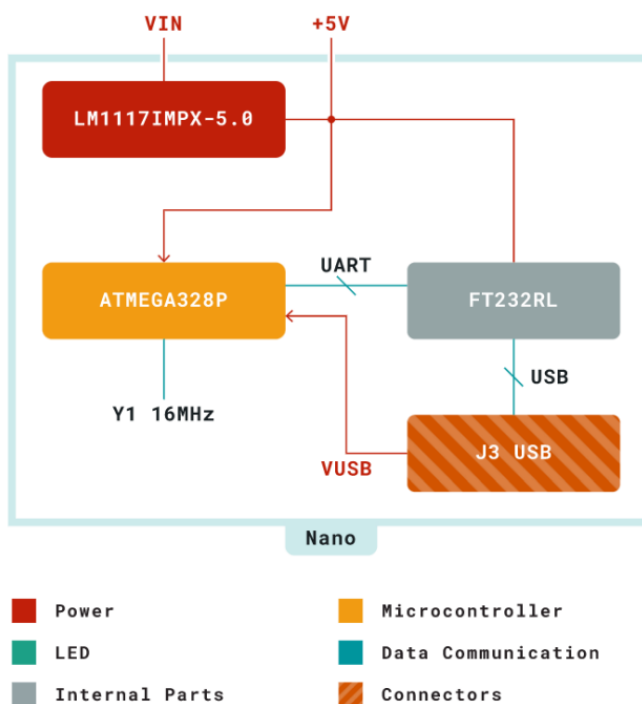


Рис. 3.3 – Блок-схема Arduino Nano

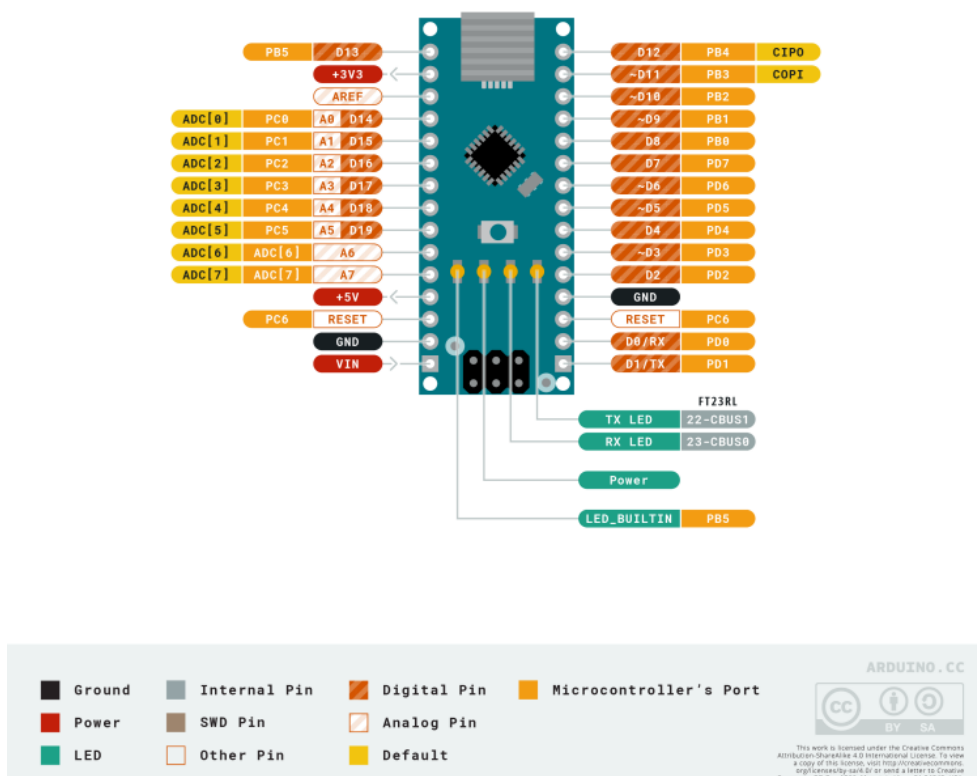


Рис. 3.4 - Виводи Arduino Nano

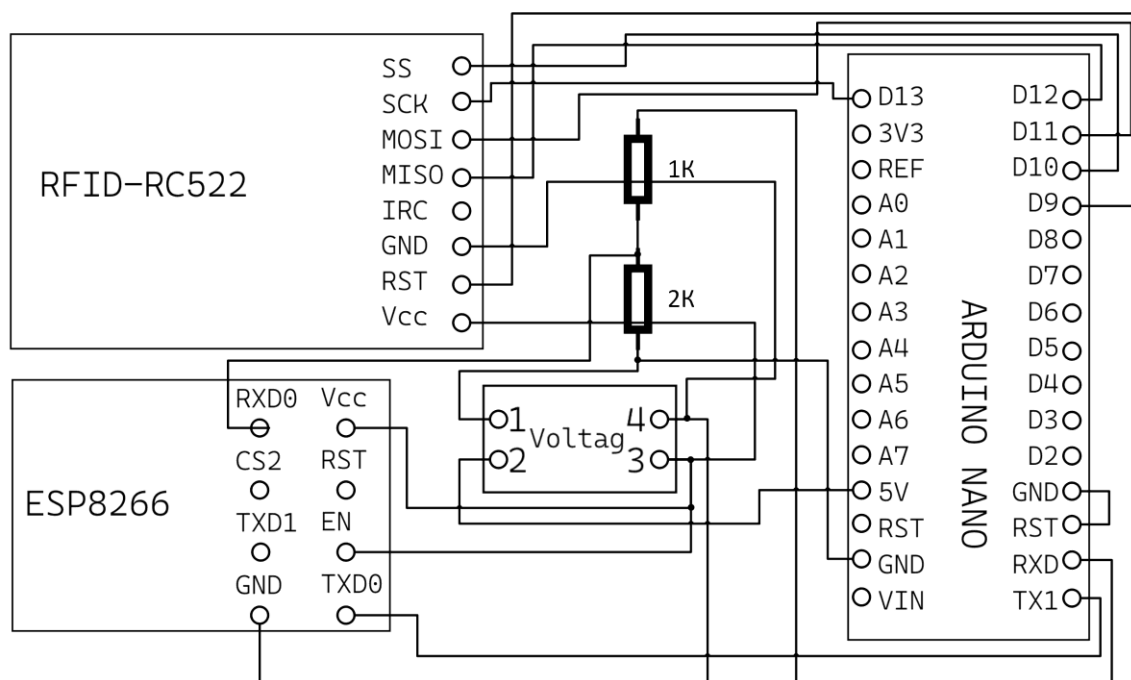


Рис. 3.5 – Схема підключення ESP8266 та RFID-RC522 до Arduino Nano

Arduino IDE – це інтегроване середовище розробки для Windows, MacOS та Linux, розроблене на C та C++, призначене для створення та завантаження програм на Arduino-сумісні плати, а також на плати інших виробників.

Просте та функціональне середовище розробки для створення власного ПЗ, яким керуються численні пристрої, зібрані початківцями та досвідченими електроніками. З'єднання ПК із мікроконтролером реалізовано через інтерфейс USB. Код мовою C та C++ пишеться в редакторі, в якому є підсвічування команд.

Середовище розробки програмного забезпечення (IDE) для конструкторів на основі плат Arduino з'явилося одночасно з апаратною частиною популярної платформи.

Низький поріг входження у конструювання своїх електронних пристроїв досягається з допомогою простоти написання програмної частини. У кожен мікроконтролер вшитий завантажувач, який трансформує інструкції, написані людиною, в машинні коди без використання апаратного програматора. Розробники-початківці та досвідчені інженери-електроніки

цінують легкість, з якою створюється працюючий прототип або готовий пристрій. Для роботи з апаратурою потрібен лише комп'ютер із встановленим середовищем розробки, яке вільно поширюється.

Arduino IDE включає не тільки редактор, але ще й компілятор, що працює в парі з завантажувачем. Для ознайомлення з можливостями передбачені готові шаблони коду, які одразу завантажуються у пристрій. Початківцям зазвичай демонструють миготіння світлодіода із заданою періодичністю. Шаблон цього коду можна завантажити в редактор через меню "Файл", в якому є окремий пункт "Приклади" або, залежно від версії, "Зразки".

Середовище розробки розповсюджується вільно. Інсталювати Arduino IDE можна завантажити з нашого сайту для операційних систем Windows, Mac OS, Linux. Як компілятор середовище використовує відомий `avr-gcc`, а значить, що «з коробки» користувач отримує підтримку оригінальних плат з архітектурою AVR та аналогів.

Середовище розробки не призначене для ведення складних проектів з деревоподібною структурою файлів, але воно, як і раніше, залишається популярним, оскільки робота з апаратною частиною не передбачає розгалужень. Оболонка IDE написана на Java і дуже важка при запуску і компіляції коду, але головне завдання виконує. Надіслати код до мікроконтролера за допомогою середовища розробки можна легко та швидко[48].

Для прошивки необхідно додатково налаштувати програму Arduino IDE. Зайти в налаштування: File – Preferences. У поле Additional Boards Manager URLs необхідно вставити посилання на Arduino Core для ESP8266.

Після прошивки необхідно додати декілька бібліотек для коректної роботи розробленого коду для Arduino: `MFRC522.h` та `ESP8266WiFi.h`. Це забезпечить роботу коду та дозволить забезпечити зв'язок через Wi-Fi мережу.

Для перевірки наявності RFID-мітки у базі даних використовується MySQL.

База MySQL – це реляційна основа. Так називаються бази, організовані в виді пов'язаних між собою таблиць. Ви можете змінити відкритий вихідний код. Ви можете самостійно встановити програму. Якщо вивчити вільний код, то можна налаштувати його під свої потреби. Коли потрібна розширена підтримка, тоді необхідно придбати ліцензовану версію.

Комп'ютери, які запускають програму СУРБД, називаються клієнтами. Для отримання даних вони підключаються до сервера. MySQL сервер – це багатопотоковий обслуговуючий пристрій для роботи з базами даних. Така система і називається «клієнт-сервер».

У середовищі MySQL відбуваються такі процеси:

- створюється база даних MySQL, визначається відношення таблиці;
- клієнти роблять запити, використовуючи команди SQL;
- сервер відповідає запитаній інформації, яка надсилається клієнту.

Популярними інтерфейсами є SequelPro, DBVisualizer, MySQL WorkBench та Navicat DB Admin Tool. Серед них є безкоштовні і ті, якими можна скористатися на комерційних умовах. Багато інтерфейсів сумісні з популярними операційними системами. Деякі працюють лише з macOS.

SQL - мова структурованих запитів. Він використовується для зв'язку клієнтських комп'ютерів із сервером. Синтаксис SQL використовується і в його різновидах (PostgreSQL і Microsoft SQL). Саме він завжди використовується як основна мова для зв'язку з базою даних.

MySQL — це реляційна система керування базами даних (RDBMS).

Деякі функції MySQL включають:

- Система управління реляційною базою даних (RDBMS), яка може зберігати дані в таблицях і виконувати операції з цими даними за допомогою SQL.
- Підтримує кілька механізмів зберігання, включаючи InnoDB, який дозволяє транзакції та блокування на рівні рядків.
- Пропонує низку функцій, яких немає в інших RDBMS, таких як підтримка збережених процедур і тригерів.
- Може використовуватися для роботи як малих, так і великих веб-сайтів.
- Програмне забезпечення з відкритим вихідним кодом, випущене за Загальною публічною ліцензією GNU, що означає, що воно вільне для використання[54].

Для створення бази даних використовуються наступні команди:

```
// Створення бази даних з назвою «dtb_uid»
```

```
CREATE DATABASE dtb_uid;
```

```
// Вхід в базу даних
```

```
USE dtb_uid;
```

```
// Створення таблиці з назвою «rfid_table», з полями UID (символьний тип, 24 символа, код с RFID-мітки), FirstName (символьний тип, 20 символів, ім'я), LastName (символьний тип, 20 символів, прізвище), JobTitle (символьний тип, 20 символів, посада).
```

```
CREATE TABLE rfid_table
```

```
(
```

```
  uid VARCHAR(24),
```

```
  FirstName VARCHAR(20),
```

```
  LastName VARCHAR(20),
```

```
  JobTitle VARCHAR(20). );
```


Щоб перевірити, чи є мітка у базі даних, необхідно виконати SQL-запит:

```
SELECT * FROM rfid_table WHERE uid = " + uid + "
```

Для виконання запиту серверу спочатку необхідно отримати дані для перевірки.

3.2. Опис процесу роботи функцій програми для комп'ютера

Бібліотека WiFi.h - це бібліотека для Arduino, яка дозволяє використовувати WiFi shield для підключення до бездротових мереж і інтернету. Вона базується на коді Arduino LLC і створена у 2011 році. Вона дозволяє налаштовувати WiFi параметри, сканувати доступні мережі, створювати WiFi клієнтів і серверів, використовувати UDP протокол і багато іншого[51].

Бібліотека mysqlcppconn.h - це бібліотека для C++ додатків, які комунікують з MySQL серверами баз даних. Вона є частиною MySQL Connector/C++, який реалізує три різні API для роботи з MySQL:

- Новий X DevAPI для додатків, написаних на C++11. Цей API дозволяє працювати з документним сховищем MySQL і виконувати традиційні SQL запити.
- Новий X DevAPI для C для додатків, написаних на чистому C. Цей API також дозволяє працювати з документним сховищем MySQL і виконувати традиційні SQL запити.
- Старий JDBC4-базований API, який також реалізований у версії 1.1 коннектора. Цей API дозволяє виконувати тільки традиційні SQL запити.
- Бібліотека mysqlcppconn.h містить функції і класи для підключення до MySQL сервера, створення та виконання запитів,

отримання та обробки результатів, керування транзакціями, обробки помилок і винятків і багато іншого[52].

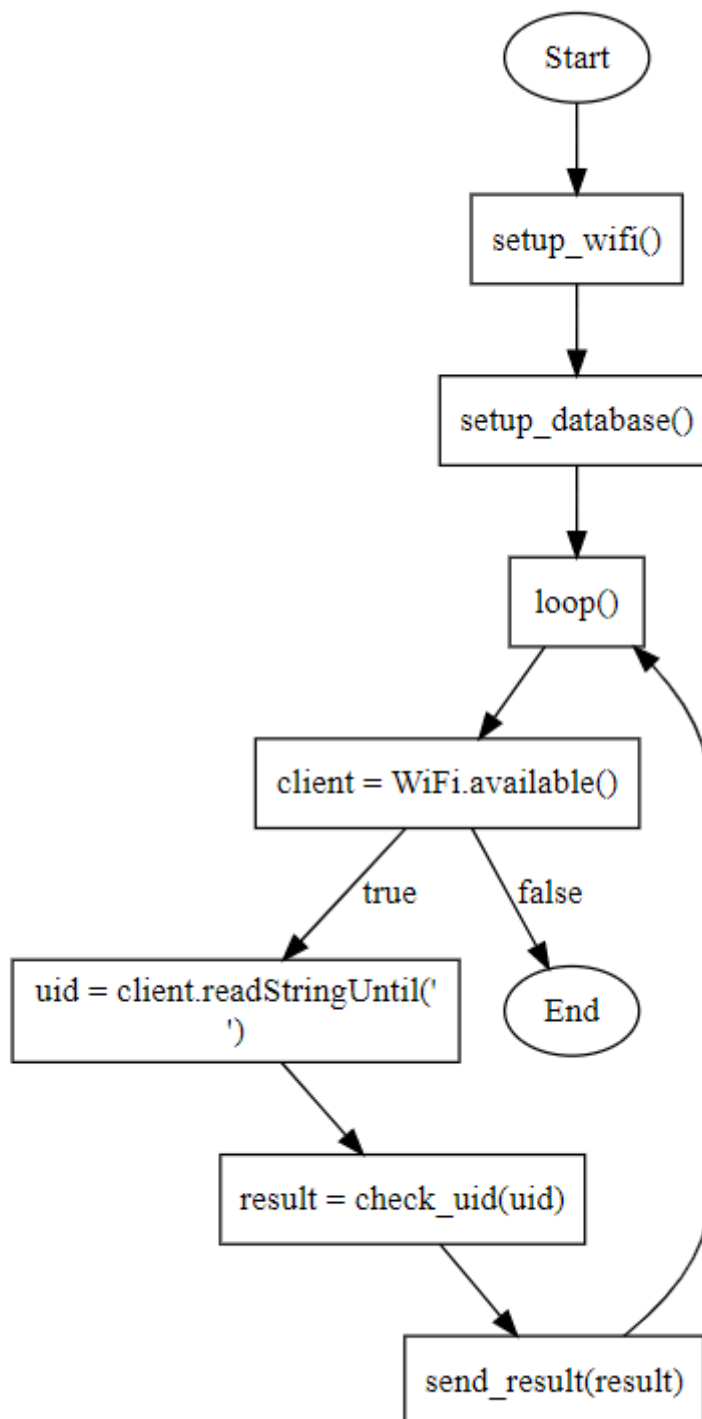


Рис. 3.1 – Блок-схема алгоритму роботи сервера

`void setup_wifi` - функція для ініціалізації Wi-Fi мережі, що спочатку виводить повідомлення про те, до якої мережі підключається, передає задані параметри назви мережі та паролю до неї. До тих пір, поки з'єднання не встановиться, відправляється крапка з затримкою в 500 мілісекунд. Якщо підключення відбулося, виводиться повідомлення про це та ір-адреса.

`void setup_database` - функція для ініціалізації бази даних `mysql`, у якій перевіряється наявність RFID-мітки у заданій базі через програму. Спочатку виводиться повідомлення, до якої бази даних йде підключення. Далі створюється об'єкт драйверу для роботи з базою даних, йде підключення з заданими параметрами. Обирається необхідна база даних із списку. Виводиться повідомлення про вдале підключення.

Якщо підключення не вдається, виводиться повідомлення про це та помилка.

`bool check_uid(String uid)` - функція перевірки наявності мітки RFID у базі даних. Створюється об'єкт для виконання SQL-запитів до бази даних. Формується необхідний для пошуку запит. Далі - виокнується, Якщо є результат, це означає, що така мітка є в базі, тому проходить допуск. В іншому випадку функція повертає логічний нуль. Звільняє пам'ять створених об'єктів.

Якщо виконати запит не вдається, виводиться відповідне повідомлення і помилка.

`void send_result(bool result)` - функція для відправки відповіді від бази даних на Arduino через Wi-Fi з'єднання.

`void setup()` - викликаються функції ініціалізації, налаштовується швидкість передачі даних.

`void loop()` - якщо клієнт доступний, відбувається зчитування RFID-мітки з буфера обміну і збереження в змінну. Далі виводиться `uid` мітки, викликається функція перевірки і виводиться результат, потім - відправляється на Arduino.

```
// Підключення бібліотек WiFi та mysqlcppconn
#include <WiFi.h>
#include <mysqlcppconn.h>

// Встановлення констант для підключення до WiFi та бази даних
const char* ssid = "your_ssid";
const char* password = "your_password";
const char* db_host = "your_db_host";
const char* db_user = "your_db_user";
const char* db_pass = "your_db_pass";
const char* db_name = "your_db_name";

// Створення об'єктів WiFiClient та sql::Connection
WiFiClient client;
sql::Connection* conn;

// Оголошення змінних для зберігання uid та результату перевірки
String uid;
bool result;

// Функція для налаштування WiFi
void setup_wifi() {
  // Виведення інформації про підключення до WiFi
  Serial.print("Connecting to ");
  Serial.println(ssid);

  // Початок підключення до WiFi з вказаним ssid та паролем
  WiFi.begin(ssid, password);
```

```
// Очікування, поки WiFi не підключиться
while (WiFi.status() != WL_CONNECTED) {
    // Затримка на 500 мс
    delay(500);
    // Виведення крапки для індикації процесу підключення
    Serial.print(".");
}
// Виведення інформації про успішне підключення до WiFi
Serial.println("");
Serial.println("WiFi connected");
// Виведення IP-адреси пристрою
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

// Функція для налаштування бази даних
void setup_database() {
    // Виведення інформації про підключення до бази даних
    Serial.print("Connecting to ");
    Serial.println(db_host);

    // Спроба підключитися до бази даних за допомогою sql::Driver
    try {
        // Отримання екземпляру sql::Driver
        sql::Driver* driver = get_driver_instance();
        // Створення об'єкта sql::Connection з вказаними параметрами
        conn = driver->connect(db_host, db_user, db_pass);
        // Встановлення схеми бази даних
        conn->setSchema(db_name);
    }
```

```

// Виведення інформації про успішне підключення до бази даних
Serial.println("Database connected");
}
// Ловлення винятку sql::SQLException у разі невдачі
catch (sql::SQLException& e) {
    // Виведення інформації про невдале підключення до бази даних
    Serial.println("Database connection failed");
    // Виведення повідомлення про помилку
    Serial.println(e.what());
}
}

// Функція для перевірки uid у базі даних
bool check_uid(String uid) {
    // Спроба виконати SQL-запит до бази даних
    try {
        // Створення об'єкта sql::Statement для виконання SQL-запитів
        sql::Statement* stmt = conn->createStatement();

        // Формування SQL-запиту для вибірки з таблиці rfid_table за uid
        String query = "SELECT * FROM rfid_table WHERE uid = " + uid + """;

        // Виконання SQL-запиту та отримання результату у вигляді
sql::ResultSet
        sql::ResultSet* res = stmt->executeQuery(query.c_str());

        // Перевірка, чи є якийсь рядок у результаті
        if (res->next()) {

```

```
// Якщо так, то повернути true
return true;
}
else {
    // Якщо ні, то повернути false
    return false;
}

// Звільнення пам'яті, виділеної для sql::Statement та sql::ResultSet
delete stmt;
delete res;
}

// Ловлення винятку sql::SQLException у разі невдачі
catch (sql::SQLException& e) {
    // Виведення інформації про невдаче виконання SQL-запиту
    Serial.println("SQL query failed");
    // Виведення повідомлення про помилку
    Serial.println(e.what());

    // Повернення false
    return false;
}
}

// Функція для надсилання результату перевірки до клієнта
void send_result(bool result) {
    // Перевірка, чи результат дорівнює true
    if (result) {
        // Якщо так, то надіслати символ '1' до клієнта
```

```
    client.write('1');
}
else {
    // Якщо ні, то надіслати символ '0' до клієнта
    client.write('0');
}
}

// Функція для налаштування програми
void setup() {
    // Встановлення швидкості передачі даних для серійного порту
    Serial.begin(9600);

    // Виклик функції для налаштування WiFi
    setup_wifi();

    // Виклик функції для налаштування бази даних
    setup_database();
}

// Функція для повторювання програми
void loop() {
    // Перевірка, чи є доступний клієнт WiFi
    if (client = WiFi.available()){
        // Якщо так, то прочитати uid з клієнта до символу '\n'
        uid = client.readStringUntil('\n');

        // Вивести uid на серійний порт
        Serial.print("UID:");
```



```
Serial.println(uid);

// Виклик функції для перевірки uid у базі даних та збереження
результату
result = check_uid(uid);

// Вивести результат на серійний порт
Serial.print("Result:");
Serial.println(result);

// Виклик функції для надсилання результату до клієнта
send_result(result);
}}
```

Представлена програма компілюється за допомогою середовища розробки Visual Studio 2022.

3.3. Опис процесу роботи функцій програми для Arduino

На рис. 3.2 показано блок-схему алгоритму роботи мікроконтролера, що входить до складу платформи Arduino Nano.

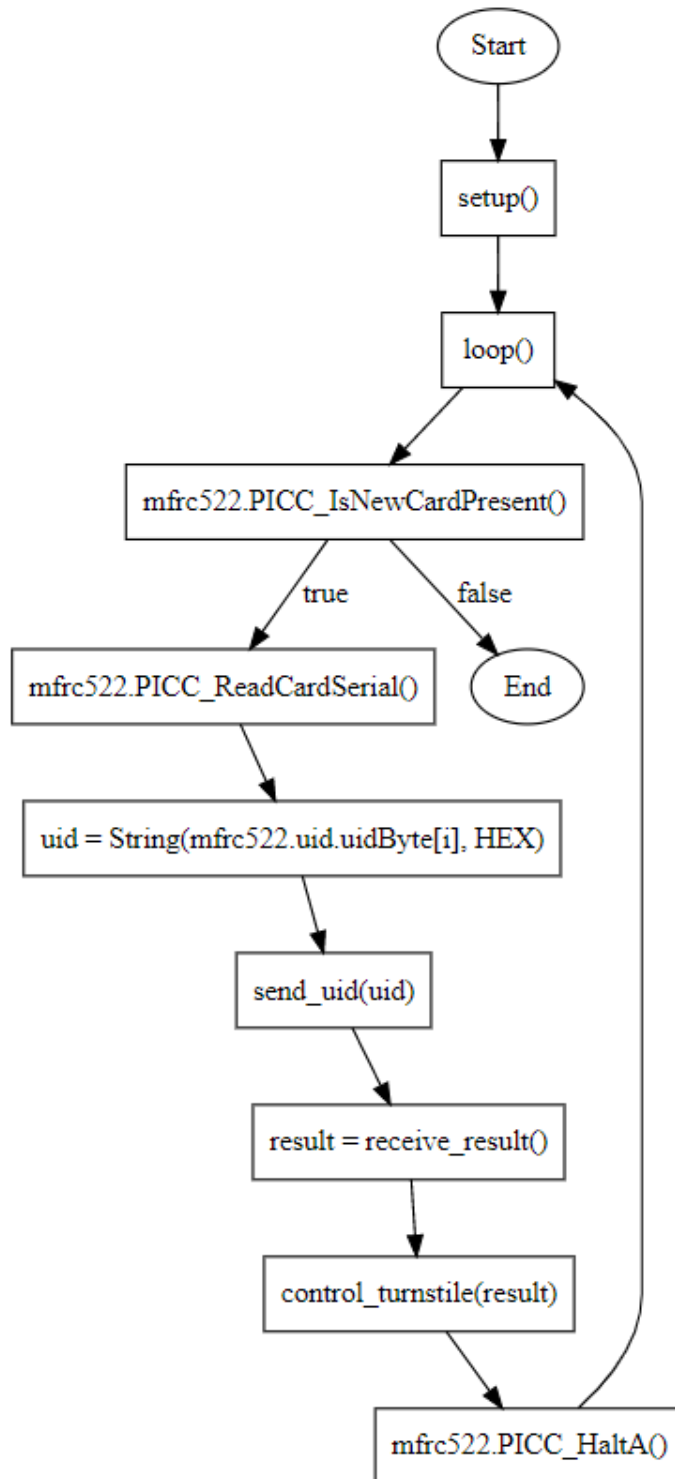


Рис. 3.2 – Блок-схема алгоритму роботи мікроконтролера

Програма для реалізації запропонованого алгоритму показана у наступному лістингу:

```
// Підключення бібліотек MFRC522, Servo та ESP8266WiFi
#include <MFRC522.h>
#include <Servo.h>
#include <ESP8266WiFi.h>

// Встановлення констант для підключення до WiFi та комп'ютера
const char* ssid = "your_ssid";
const char* password = "your_password";
const char* host = "your_host";
const int port = 80;

// Встановлення констант для пінів RST та SS модуля MFRC522
#define RST_PIN 9
#define SS_PIN 10

// Створення об'єкта MFRC522 з вказаними пінами
MFRC522 mfrc522(SS_PIN, RST_PIN);

// Створення об'єкта WiFiClient для зв'язку з комп'ютером
WiFiClient client;

// Оголошення змінної для зберігання uid карти
String uid;

// Оголошення змінної для зберігання результату перевірки карти
char result;

// Створення об'єкта Servo для керування сервоприводом
Servo servo;

// Функція для налаштування програми
```

```
void setup() {  
    // Встановлення швидкості передачі даних для серійного порту  
    Serial.begin(9600);  
  
    // Початок роботи SPI  
    SPI.begin();  
  
    // Ініціалізація модуля MFRC522  
    mfrc522.PCD_Init();  
  
    // Виведення інформації про готовність до сканування RFID-карт  
    Serial.println("Ready to scan RFID cards");  
    // Встановлення констант для пінів реле та сервоприводу  
    #define RELAY_PIN 4  
    #define SERVO_PIN 3  
    // Встановлення констант для кутів відкриття та закриття  
сервоприводу  
    #define OPEN_ANGLE 90  
    #define CLOSE_ANGLE 0  
  
    // Початок підключення до WiFi з вказаним ssid та паролем  
    WiFi.begin(ssid, password);  
    // Очікування, поки WiFi не підключиться  
    while (WiFi.status() != WL_CONNECTED) {  
        // Затримка на 500 мс  
        delay(500);  
        // Виведення крапки для індикації процесу підключення  
        Serial.print(".");  
    }  
}
```

```
// Виведення інформації про успішне підключення до WiFi
Serial.println("");
Serial.println("WiFi connected");
// Виведення IP-адреси пристрою
Serial.println("IP address:");
Serial.println(WiFi.localIP());
}

// Функція для повторювання програми
void loop() {
    // Перевірка, чи є нова карта, присутня біля модуля MFRC522
    if(mfrc522.PICC_IsNewCardPresent()) {
        // Перевірка, чи можна прочитати серійний номер карти
        if(mfrc522.PICC_ReadCardSerial()) {
            // Очищення змінної uid
            uid = "";
            // Формування рядка uid з байтів серійного номера карти
            for (byte i = 0; i < mfrc522.uid.size; i++) {
                uid += String(mfrc522.uid.uidByte[i], HEX);
            }
            // Виведення uid на серійний порт
            Serial.print("UID:");
            Serial.println(uid);

            // Виклик функції для надсилання uid до комп'ютера
            send_uid(uid);

            // Виклик функції для отримання результату перевірки карти від
            комп'ютера та збереження його
```

```
result = receive_result();

// Виведення результату на серійний порт
Serial.print("Result: ");
Serial.println(result);

// Виклик функції для керування турнікетом в залежності від
результату
control_turnstile(result);
}

// Зупинка роботи з картою
mfrc522.PICC_HaltA();
}
}

// Функція для надсилання uid до комп'ютера
void send_uid(String uid) {
    // Перевірка, чи можна підключитися до комп'ютера за вказаним
хостом та портом
    if (client.connect(host, port)) {
        // Виведення інформації про успішне підключення до комп'ютера
        Serial.println("Connected to computer");
        // Надсилання uid до комп'ютера
        client.println(uid);
        // Завершення зв'язку з комп'ютером
        client.stop();
    }
    else {
```

```
// Виведення інформації про невдале підключення до комп'ютера
Serial.println("Connection to computer failed");
}
}

// Функція для отримання результату перевірки карти від комп'ютера
char receive_result() {
    // Перевірка, чи можна підключитися до комп'ютера за вказаним
хостом та портом
    if (client.connect(host, port)) {
        // Виведення інформації про успішне підключення до комп'ютера
        Serial.println("Connected to computer");
        // Очікування, поки комп'ютер не надішле якісь дані
        while (!client.available()) {
            // Затримка на 1 мс
            delay(1);
        }
        // Повернення символу, отриманого від комп'ютера
        return client.read();
    }
    else {
        // Виведення інформації про невдале підключення до комп'ютера
        Serial.println("Connection to computer failed");
        // Повернення символу '0'
        return '0';
    }
}

// Функція для налаштування турнікету
```

```
void setup_turnstile() {  
    // Встановлення режиму виводу для піна реле  
    pinMode(RELAY_PIN, OUTPUT);  
    // Встановлення низького рівня сигналу для піна реле  
    digitalWrite(RELAY_PIN, LOW);  
    // Приєднання сервоприводу до піна SERVO_PIN  
    servo.attach(SERVO_PIN);  
    // Встановлення кута закриття для сервоприводу  
    servo.write(CLOSE_ANGLE);  
}  
  
// Функція для керування турнікетом в залежності від результату  
void control_turnstile(char result) {  
    // Перевірка, чи результат дорівнює '1'  
    if (result == '1') {  
        // Якщо так, то встановити високий рівень сигналу для піна реле  
        digitalWrite(RELAY_PIN, HIGH);  
        // Встановити кут відкриття для сервоприводу  
        servo.write(OPEN_ANGLE);  
        // Вивести повідомлення про відкриття турнікету на серійний порт  
        Serial.println("Turnstile opened");  
    }  
    // Якщо ні, то  
    else {  
        // Встановити низький рівень сигналу для піна реле  
        digitalWrite(RELAY_PIN, LOW);  
        // Встановити кут закриття для сервоприводу  
        servo.write(CLOSE_ANGLE);  
        // Вивести повідомлення про закриття турнікету на серійний порт
```



```
Serial.println("Turnstile closed");
}}
```

Спочатку відбувається підключення бібліотек для роботи с RFID модулем RC522 та Wi-Fi модулем ESP8226.

Задаються параметри підключення до Wi-Fi мережі. У змінну ssid необхідно ввести ім'я мережі; у password - пароль.

Задаються параметри для підключення до комп'ютера посту охорони: IP-адреса та порт для обміну даними.

Задаються піни для підключення RFID модуля RC522 до Arduino по SPI протоколу. RST_PIN - для скидання модуля, SS_PIN для вибору пристрою.

Створюється об'єкти для роботи з модулями RFID та підключення до Wi-Fi-мережі, змінні для мітки та для зберігання відповіді від комп'ютера.

У void setup() налаштовується швидкість передачі даних по порту; ініціалізується протокол SPI для передачі даних з RFID-модулем і безпосередньо сам модуль. Виводиться повідомлення про готовність до роботи. Далі йде підключення до Wi-Fi мережі за заданими параметрами - і очікується, доки підключення не буде успішним. Після підключення виводиться повідомлення про це.

void loop() - виконується перевірка наявності RFID-мітки в полі дії модуля. Якщо мітка наявна, збирається UID RFID-мітки із байтів в рядок, зберігається у змінну uid. Мітка виводиться на екран, відправляється на комп'ютер. Отримується відповідь від комп'ютера у вигляді логічної одиниці чи нуля, виводиться на екран; в залежності від відповіді турнікет відчиняється або ні. Поточна мітка закривається.

void send_uid(String uid) - відправляє UID RFID-мітки на комп'ютер. Спроба підключитися до комп'ютера по заданим параметрам; вивід повідомлення в залежності від результату. Дані відправляються командою client.println(uid), з'єднання закривається.

`char receive_result()` - отримання відповіді від комп'ютера. Відбувається підключення до комп'ютера за заданими параметрами, виводиться повідомлення. Очікування відповіді від комп'ютера. Зчитування відповіді від комп'ютера з буферу клієнта, переформатування у символний тип.

`void setup_turnstile()` - ініціалізація сервопривода для керування турнікетом. Встановлення турнікета у низький стан, закрите положення.

`void control_turnstile(char result)` - в залежності від результату відкриває турнікет або не відкриває. Реле перемикається у високий стан - а сервопривод у відкрите положення.

Представлена програма компілюється за допомогою середовища розробки Arduino IDE та завантажується у мікроконтролер через USB інтерфейс.

Висновки до розділу 3

1. Описано принцип передачі даних між компонентами системи.
2. Наведено інформацію про використовувані інструменти для створення програмного забезпечення.
3. Розписані принципи роботи функцій, що використовуються у програмах для комп'ютера посту охорони та для модуля Arduino.
4. Створено програмне забезпечення для Arduino для турнікету пропускної систем та для комп'ютера на мові C++ для прийняття даних через мережу Wi-Fi і перевірки наявності мітки у базі даних.

ВИСНОВКИ

1. Виконано аналітичний огляд структури та принципу роботи існуючих пропускних систем.
2. Розроблено структуру пропускної системи .
3. Розроблено програмне забезпечення для віддаленого керування пропускною системою, що складається з серверної та клієнтської частин.
4. Серверна частина програмного забезпечення прийняття забезпечує збереження та пошук даних у базі даних MySQL.
5. Клієнтська частина програмного забезпечення забезпечує зчитування даних RFID-міток, їх відправку через Wi-Fi з'єднання та керування турнікетом.
6. Для стабільної роботи системи рекомендується підключати не більше чотирьох турнікетів, але можливе розширення та вдосконалення за потреби.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система - Вікіпедія. URL: <https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0> (дата звернення: 15.03.2023).
2. Системи доступу. URL: <https://tsm.ua/uk/sistemi-dostupu.html> (дата звернення: 16.03.2023).
3. Буров Є.В. Комп'ютерні мережі: підруч. - Львів: Магнолія плюс, 2006. – 264 с.
4. Системи контролю доступу Київ. URL: <https://ohholding.com.ua/uk/tekhnicheskie-pravila-bezopasnosti/sistema-kontrolya-i-upravleniya-dostupom> (дата звернення: 18.03.2023).
5. Рамський Ю.С., Олексюк В.П., Балик А.В. Адміністрування комп'ютерних мереж і систем: Навч. пос. – Тернопіль: Навчальна книга – Богдан, 2010. – 196 с
6. Електронна ідентифікація. URL: <https://id.gov.ua/> (дата звернення: 18.03.2023).
7. Система контролю і управління доступом: опис, які вирішує завдання, компоненти. URL: <https://deps.ua/ua/knowegable-base/reference-information/7824.html> (дата звернення: 20.03.2023).
8. Р. Седжвик Алгоритми на C++ // К., 2019, ISBN: 978-5-907144-21-7
9. Е. Таненбаум Комп'ютерні мережі // К., 2021, ISBN — 978-5-4461-1248-7
10. Контроль доступу (інформатика). URL: [https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D1%83_\(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0\)](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D1%83_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) (дата звернення: 20.03.2023).

11. Автоматизована система контролю доступу. URL: https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8E_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D1%83 (дата звернення: 21.03.2023).
12. Контроль системи доступу. URL: <https://ip24.com.ua/ua/g43575972-komplekt-sistemy-kontrolya> (дата звернення: 21.03.2023).
13. Programming Arduino Next Steps: Going Further with Sketches / Simon Monk. McGraw Hill TAB; 2nd edition, 2017. 272 p.
14. Andrew S. Tanenbaum. Computer Networks / Andrew S. Tanenbaum David J. Wetherall – 5th Edition : Prentice Hall, 2011. – 900 p.
15. Турнікети. URL: <https://rozetka.com.ua/ua/turniketi/c4650483/> (дата звернення: 21.03.2023).
16. Мікроконтролер. URL: <https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%B%D0%B5%D1%80> (дата звернення: 22.03.2023).
17. Пристрій керування. URL: https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D1%81%D1%82%D1%80%D1%96%D0%B9_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення: 22.03.2023).
18. Класифікація і структура мікроконтролерів. URL: https://ua-referat.com/%D0%9A%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F_%D1%96_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BC%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%B

- E%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80%D1%96%D0%B2 (дата звернення: 22.03.2023).
19. ID Card - Обладнання для RFID ідентифікації в Україні. URL: <https://idcard.com.ua/ua/> (дата звернення: 01.04.2023).
 20. Що таке система RFID, в чому її особливості використання. URL: <https://idcard.com.ua/ua/blog/chto-takoe-sistema-rfid-v-chem-ee-osobennosti-ispolzovaniya/> (дата звернення: 01.04.2023).
 21. Абрамов В.О. Базові технології комп'ютерних мереж: навч. посіб. / В.О. Абрамов, СЮ. Клименко. - К.: Київ, ун-т ім. Б. Грінченка, 2011. - 291 с.
 22. RFID модули. URL: <https://prom.ua/Rfid-moduli.html> (дата звернення: 02.04.2023).
 23. Що таке система RFID, в чому її особливості використання. URL: <https://idcard.com.ua/ua/blog/chto-takoe-sistema-rfid-v-chem-ee-osobennosti-ispolzovaniya/> (дата звернення: 02.04.2023).
 24. Про платформу Arduino. URL: <https://doc.arduino.ua/ru/about/> (дата звернення: 02.04.2023).
 25. Пост охорони на підприємстві. URL: <https://www.ohrana.com.ua/blog/post-ohoroni-na-pidpriemstvi/> (дата звернення: 02.04.2023).
 26. Сервер. URL: <https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80> (дата звернення: 05.04.2023).
 27. Що таке сервер бази даних. URL: <https://www.hostinger.ua/posibnyky/shho-take-server-bazy-danyh/> (дата звернення: 05.04.2023).
 28. Типи баз даних. URL: <https://www.ukraine.com.ua/uk/blog/database-server-types.html> (дата звернення: 05.04.2023).

29. Що таке база даних. URL: <https://aperep.kpi.ua/shco-take-basa-danykh> (дата звернення: 08.04.2023).
30. Основні відомості про бази даних. URL: [ua/office/%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%96-%D0%B2%D1%96%D0%B4%D0%BE%D0%BC%D0%BE%D1%81%D1%82%D1%96-%D0%BF%D1%80%D0%BE-%D0%B1%D0%B0%D0%B7%D0%B8-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85-a849ac16-07c7-4a31-9948-3c8c94a7c204](https://ua.office.com/office/%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%96-%D0%B2%D1%96%D0%B4%D0%BE%D0%BC%D0%BE%D1%81%D1%82%D1%96-%D0%BF%D1%80%D0%BE-%D0%B1%D0%B0%D0%B7%D0%B8-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85-a849ac16-07c7-4a31-9948-3c8c94a7c204) (дата звернення: 08.04.2023).
31. RFID модуль RC522 з Arduino. URL: <https://www.robostore.com.ua/ua/rfid-modul-rc522-with-arduino/> (дата звернення: 15.04.2023).
32. Mini RFID RC522 модуль. URL: <https://www.arduinka.top/product/mini-rfid-rc522-modul> (дата звернення: 15.04.2023).
33. Радіодеталі. Інтернет-крамниця.. URL: <https://radiodetali.com.ua/ua/catalog/mikroschemy> (дата звернення: 15.04.2023).
34. Мікросхеми, види, призначення. URL: <https://radiodetali.com.ua/stati/mikroschemy-vidy-naznachenie> (дата звернення: 15.04.2023).
35. Знайомимось з модулем ESP8266. URL: <https://hobbytech.com.ua/%D0%B7%D0%BD%D0%B0%D0%BA%D0%BE%D0%BC%D0%B8%D0%BC%D1%81%D1%8F-%D1%81-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B5%D0%BC-esp8266-%D0%BF%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D0%B5%D0%B5/> (дата звернення: 15.04.2023).

36. Мікросхема. URL:
<https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D1%81%D1%85%D0%B5%D0%BC%D0%B0> (дата звернення: 15.04.2023).
37. Мікроелектроніка. URL:
<https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D1%96%D0%BA%D0%B0> (дата звернення: 16.04.2023).
38. MFRC522 Standard performance MIFARE and NTAG frontend. URL:
<https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf> (дата звернення: 16.04.2023).
39. Andrew S. Tanenbaum. Computer Networks / Andrew S. Tanenbaum David J.Wetherall – 5th Edition : Prentice Hall, 2011. – 900 p.
40. RFID модуль с карточкою доступа для Arduino RC522. URL:
<https://www.robostore.com.ua/moduli-i-datchiki/besprovodnye-datchiki/rfid-modul-rc522-s-kartochnoj-dostupa/> (дата звернення: 16.04.2023).
41. Програмне забезпечення. URL:
https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F (дата звернення: 16.04.2023).
42. Класифікація ПЗ. URL:
<https://sites.google.com/site/inform323/klasifikacia-pz> (дата звернення: 16.04.2023).
43. Мова програмування. URL:
https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83

- %D0%B2%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення: 23.04.2023).
44. Огляд і основи мови програмування C++. URL: http://www.znannya.org/?view=Cpp_basics (дата звернення: 23.04.2023).
45. Що таке інтегроване середовище розробки (IDE). URL: <https://soccerspen.com/uk/what-is-integrated-development-environment-ide/> (дата звернення: 23.04.2023).
46. Programming Arduino Next Steps: Going Further with Sketches / Simon Monk. McGraw Hill TAB; 2nd edition, 2017. 272 с.
47. RFID — технологія автоматичної ідентифікації. URL: https://www.vostok.dp.ua/ukr/infa1/rfid/rfid_tekhnologiya/ (дата звернення: 23.04.2023).
48. Arduino-ide. URL: <https://arduino-ide.com/> (дата звернення: 28.04.2023).
49. VisualStudio. URL: <https://visualstudio.microsoft.com/> (дата звернення: 28.04.2023).
50. Understanding Wi-Fi and How It Works. URL: <https://www.lifewire.com/what-is-wi-fi-2377430> (дата звернення: 28.04.2023).
51. Arduino-esp32. URL: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFi.h> (дата звернення: 28.04.2023).
52. MySQL Connector/C++ Documentation. URL: <https://dev.mysql.com/doc/dev/connector-cpp/8.0/> (дата звернення: 28.04.2023).
53. Що таке MYSQL. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-mysql/> (дата звернення: 28.04.2023).
54. MySQL. URL: <https://www.mysql.com/> (дата звернення: 28.04.2023).

ДОДАТОК

ПУБЛІКАЦІЯ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

УДК 004.7

ПРИСТРІЙ ВІДДАЛЕНОГО КЕРУВАННЯ ПРОПУСКНОЮ СИСТЕМОЮ

Ярмоленко А.С. – гр. БКІск-21, бакалавр, *the.dark.ferox@gmail.com*

Стаценко В.В. – д.т.н., доц., *statsenko.v@knutd.edu.ua*

Київський національний університет технологій та дизайну

Метою роботи є створення схеми взаємодії компонентів та програмного забезпечення для віддаленого керування пропускною системою.

Системи контролю доступу (СКД) широко використовуються в якості засобу запобігання несанкціонованому проникненню сторонніх осіб на територію підприємства. Також вони дозволяють забезпечити розмежування доступу співробітників в приміщення, вести автоматизований облік відвідувачів. СКД є одним зі складових елементів інтегрованих систем безпеки, до яких також входять системи відеоспостереження, охоронні сигналізації, тощо.

Більшість СКД реалізовані як електронно-механічні системи, що дозволяють перевіряти дозволи відвідувачів або співробітників на доступ до певних приміщень та у разі необхідності блокувати можливість такого доступу. З точки зору перевірки доступу такі системи поділяються на:

- автономні — інформація не передається на центральний пункт охорони і не контролюється операторами;
- централізовані (мережеві) — забезпечують обмін між центральним пунктом охорони та виконавчими пристроями;
- універсальні — можуть працювати як в автономному режимі, так і в мережевому. Якщо відбувається збій центрального пристрою управління, то перемикається на автономний режим [3].

В роботі передбачено створення СКД структурна схема якої показана на рис. 1. До її складу входять: турнікети, що забезпечують фізичне обмеження доступу; пристрій керування на базі мікроконтролера (МК) AtMega 326, що входить до складу платформи Arduino [1], комп'ютер посту охорони та сервер бази даних. Ключем, що дозволяє відвідувачам пройти

через турнікет, є картка з RFID-міткою. Коли мітку підносять до турнікету, пристрій керування зчитує інформацію з неї за допомогою RFID-модуля RC522, основою якого є мікросхема MFRC522. Модуль працює на частоті 13,56 МГц, а обмін даними з мікроконтролером здійснюється через SPI-інтерфейс. Далі МК передає інформацію з RFID-мітки на комп'ютер центрального посту охорони через Wi-Fi мережу [2]. Для цього до складу пристрою керування входить модуль ESP8266. Це дозволяє підключити до одного комп'ютера одночасно декілька турнікетів. Програма, що встановлена на комп'ютері посту охорони, шукає дані RFID-мітки у базі даних та у випадку успішного проходження перевірки надсилає до МК команду розблокування відповідного турнікету.

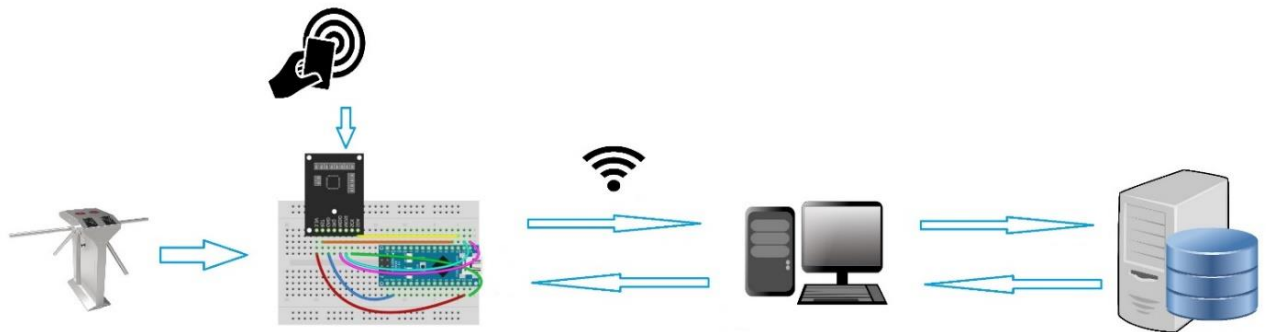


Рис. 1. Структурна схема системи контролю доступу

Запропонована система дозволяє реалізувати наступні функції:

- формування індивідуальних і групових повноважень доступу;
- облік робочого часу співробітників (перебування їх у певному приміщенні);
- облік доступу сторонніх осіб із зазначенням останнього місця їх перебування;
- контроль стану всіх приміщень (відображення плану об'єкта з коментарями в протоколі подій);
- можливість окремого блокування та розблокування будь-якого з терміналів;

Висновок. В результаті проведеного аналізу створено схему взаємодії компонентів СКД та визначено основні функції програми віддаленого керування пропускною системою.

Л і т е р а т у р а

1. Programming Arduino Next Steps: Going Further with Sketches / Simon Monk. McGraw Hill TAB; 2nd edition, 2017. 272 с.

2. Andrew S. Tanenbaum. Computer Networks / Andrew S. Tanenbaum David J. Wetherall – 5th Edition : Prentice Hall, 2011. – 900 p.

3. Системи контролю доступу [Електронний ресурс]. – Режим доступу: <https://ukrinfosystems.com.ua/uk/design-and-construction/access-control-systems>