НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Радіотехнічний факультет
Кафедра прикладної радіоелектроніки

До захисту допущено:
В.о.зав. кафедри
_____ Михайло СТЕПАНОВ
«___»_____2022 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньою програмою «Радіоз`язок і оброблення сигналів»
за спеціальністю 172 «Телекомунікації та радіотехніка»
на тему: «_Система безключового доступу_»

Виконав:
студент IV курсу, групи РА-81
_____Пінчук Андрій Володимирович_____ _____

Керівник:_____Доцент, к.т.н. Мосійчук В.С._____ _____
Посада, науковий ступінь, вчене звання,
Прізвище, ім'я, по батькові

Рецензент: доцент, к.т.н. каф. РТС Товкач Ігор Олегович _____
Посада, науковий ступінь, вчене звання,
Прізвище, ім'я, по батькові

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент (-ка) ___ _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут/факультет **радіотехнічний**

Кафедра **прикладної радіоелектроніки**

Рівень вищої освіти – **перший (бакалаврський)**

Спеціальність (спецілізація) **172 «Телекомунікації та радіотехніка»**

ЗАТВЕРДЖУЮ
В. о. зав. кафедри
_____ Михайло СТЕПАНОВ
«___»_____2022 р.

ЗАВДАННЯ
на дипломну роботу студенту
Пінчуку Андрію Володимировичу

1. **Тема роботи**: «Система безключового доступу»
керівник проекту Мосійчук Віталій Сергійович,
затверджені наказом по університету від «01» червня 2022 р. № 823-с.

2. **Строк подання студентом проекту** «20» червня 2022 року.

3. **Вихідні дані до проекту**: роз'єм для зарядки micro*USB.*; вага - не більше 1 кг;

4. **Зміст пояснювальної записки** (перелік завдань, які потрібно розробити) проаналізувати завдання; оглянути аналоги; розробити та обґрунтувати структурну та принципову схему; обрати елементну базу; підтвердити працездатність.

5. **Перелік графічного (ілюстративного) матеріалу** (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) структурна схема;

6. **Консультанти розділів роботи**[*]

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

7. **Дата видачі завдання** «02» травня 2022 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Огляд існуючих рішень	2.05.22 – 13.05.22	
2	Розробка та аналіз ТЗ	14.05.22 – 18.05.22	
3	Обґрунтування та вибір схемотехнічних рішень	19.05.22 – 6.06.22	
4	Макетування системи	7.06.22 – 12.06.22	
5	Перевірка працездатності	13.06.22 – 15.06.22	
6	Оформлення документації	16.06.20 – 20.06.22	

Студент Андрій ПІНЧУК

Керівник Віталій МОСІЙЧУК

[*] Консультантом не може бути зазначено керівника дипломної роботи(проекту)

АНОТАЦІЯ

Використання замків для захисту речей не є новим.

У наш чаш розвитку технологій можливо використовувати «розумні» замки використовуючи мікроконтролери.

За допомогою використання безконтактних зчитувачів можна позбавитися від великої кількості металевих ключів,та значно прискорити процес дистанційною видачею доступу іншим особам.

Метою дипломного проекту є створення системи безключового доступу.

Система розроблюється для наступних цілей:

- використання ID карток в якості ідентифікатора доступу.
- контроль доступу осіб до певних приміщень.
- створення рівнів доступу які мають доступ до приміщень.
- ведення запису робочого часу працівників на робочому місці.

Ключові слова: контроль доступу, безключовий доступ.

ANNOTATION

The use of locks to protect things is not new.

In our bowl of technology development, you can use "smart" locks with microcontrollers.

With the use of contactless readers you can get rid of a large number of metal keys, which will significantly speed up the process of remote access to others.

The aim of the diploma project is to create a keyless entry system.

The system is developed for the following purposes:

- use of an identification card as an access identifier.
- control of access of persons to certain premises.
- access levels that have access to the premises.
- recording the working hours of employees in the workplace.

Keywords: access control, keyless access.

Key words: access control, keyless access, arduino.

ЗМІСТ

Перелік скорочень 3
ВСТУП..... 4
1. Огляд існуючих аналогів 5
2. Обгрунтування технічних решень 19
2.1 Структурна схема 19
2.2 Вибір контролера 20
2.2.1 Raspberry Pi PICO 20
2.2.2 Arduino UNO 21
2.2.3 NodeMCU 22
2.2.4 плата FONA..... 24
2.2.5 обрана плата 25
2.3 Передавання даних 25
2.3.1 Дротові технології 26
2.3.2 Інфрачервоне випромінювання 27
2.3.3 Лазерні технології 27
2.3.4 Радіо Ethernet (IEEE 802.11) 28
2.3.5 Bluetooth 28
2.3.6 LoRa 29
2.3.7 GSM 29
2.3.8 Zeegbe 29
2.3.9 Модуль передавання даних 30
2.4. Вибір зчитувача 31
2.4.1 Сканер штрих-кодів 31
2.4.2 Модуль сканера відбитків пальців 32
2.4.3 Зчитувач ID карток 34
2.5 Джерело живлення 35
3. Макетування системи безключового доступу..... 37

3.1 Апаратна частина 37
3.2 Програмна частина 40
3.3 Перевірна працездатності 54
ВИСНОВК 55
Перелік посилань 56

ПЕРЕЛІК СКОРОЧЕНЬ

LoRa — Long Range

RBAC - Role-based access control

RFID - Radio Frequency Identification

SPI - Serial Peripheral Interface

ASCII — American Standard Code for Information Interchange

USB — Universal Serial Bus

RSSI — received signal strength indicator

IEEE — Institute of Electrical and Electronics Engineers

LAN — Local Area Network

Wifi — Wireless Fidelity

IDE — Integrated Drive Electronics

COM port — communications port

ISM — industrial, scientific and medical

ВСТУП

Контроль або авторизація доступу в найширшому сенсі існувала ще задовго до того, як люди мали активи, які варто захищати. Охорону, ворота та замки використовувалися з давніх часів для обмеження доступу людей до цінностей. Потреба в контролі доступу фактично спонукала до винаходу того, що можна вважати першою в світі безпечною обчислювальною системою.

У сучасних інформаційних технологіях авторизація стосується способів, за допомогою яких користувачі можуть отримати доступ до ресурсів у комп'ютерній системі, або неформально кажучи, «хто що може робити». Контроль доступу, є найбільш фундаментальним і найпоширенішим механізмом безпеки, який використовується сьогодні. Ця система відображається практично у всіх системах і створює великі архітектурні та адміністративні проблеми на всіх рівнях корпоративних обчислень. З точки зору бізнесу, контроль доступу може сприяти оптимальному розподілу та обміну ресурсами, але він також може розчарувати користувачів, спричинити великі адміністративні витрати та спричинити несанкціоноване розкриття або пошкодження цінної інформації.

Контроль доступу може приймати різні форми. На додаток до визначення того, чи має користувач права на використання ресурсу, система контролю доступу може обмежувати час і спосіб використання ресурсу.

Така система міститиме базу даних різних рівнів доступу та списки людей, які належать до цих рівнів доступу. Він фактично буде контролювати механізми блокування на різних дверях і бар'єрах відповідно до рівня доступу особи, яка надає облікові дані [1][2].

1. ОГЛЯД БЕЗКЛЮЧОВИХ СИСТЕМ НА РИНКУ

У цьому розділі ведеться огляд безключових систем які присутні на ринку. На базі розглянутого переліку пристроїв буде сформовано перелік характеристик, які повинен задовольняти готовий пристрій.

Система безключового доступу для авто[3]:

Рис. 1 доступ системи через радіоключ або застосунок

Пристрій безключового доступу в більшості модифікацій має стандартний набір елементів. Відмінності бувають тільки в сигналах, які посилають ретранслятор і ключ, а також принципі захисту (тільки закриває замки або ж працює разом з іммобілайзером).

Імобілайзер — пристрій, без якого неможливо запустити двигун автомобіля. Досягається це різними методами, суть яких така: без будь-яких додаткових дій, натискання кнопки брелка або піднесенням ключа до зчитувача коду, двигун запустити не вдастся[4].

Основні елементи:

1. Ключ. Варіантів виконання даного елемента маса. Це може бути звичний ключ з невеликим блоком, оснащеним кнопками. В іншому виконанні - брелок на в'язці ключів. Існують також ключ-карти. Все залежить від виробника: який дизайн і компонування він вибере для пристрою. У цьому елементі є мікросхема. Вона створює код або розшифровує сигнал, що надходить від ретранслятора. Для забезпечення максимального захисту використовується алгоритм з плаваючим кодом.
2. Антена. Цей елемент встановлюється не тільки на автомобіль, але і вмонтована в сам ключ. Одна передає сигнал, а інша приймає його. Розміри і кількість антен залежать від моделі пристрою. У більш дорогих автомобілях ці елементи встановлюються в багажнику, дверях авто і в районі торпедо. Деякі моделі систем дозволяють окремо деактивувати замок з однієї зі сторін транспорту, наприклад, якщо потрібно покласти речі в багажник, досить підійти спочатку до нього, підставити ногу під бампер, і пристрій відкриє кришку.
3. Датчики відкриття / закриття дверей. Вони потрібні для того, щоб визначити, яку функцію активувати. Ця функція дозволяє пристрою самостійно визначити, де знаходиться смарт-ключ (або зовні, або всередині авто).
4. Блок керування. Головне пристрій обробляє отримані сигнали і видає відповідну команду на замки дверей або іммобілайзер.

Рис. 2 Система обліку робочого часу IN01

Система обліку робочого часу IN01 являє собою програмно-апаратний комплекс, який включає в себе наступні компоненти [5]:

- Термінал — автономний пристрій для збору реєстрацій часу приходу/ відходу співробітників. Енергонезалежна пам'ять терміналу дозволяє зберігати такі обсяги інформації: шаблони відбитків пальців = 3 000 (оптимально 4 на одного співробітника), RFID-ідентифікатори = 10 000, події = 100 000.
- Програмне забезпечення — середовище для адміністрування терміналу і формування звітів.

Завдяки тому, що термінал оснащений незалежною пам'яттю і може працювати автономно, збереження реєстрацій не вимагає постійної зв'язку» з програмним забезпеченням (персональним комп'ютером).

Система обліку робочого часу IN01 підтримує наступні варіанти ідентифікації:

- відбиток пальця;
- RFID-ідентифікатор (карта/брелок формату Em-Marin 125 kHz);
- PIN-код.

Також можлива комбінована ідентифікація — відбиток пальця + RFID-ідентифікатор, RFID-ідентифікатор + PIN-код і т. д.

Введення в систему ідентифікатора співробітника — відбиток пальця або карта/брелок, можна реалізувати за допомогою терміналу або додаткового USB-зчитувача ZK8500.

Ціна: 307 доларів США

Розглянемо окремо контролери стаціонарних систем контролю доступу.

Рис. 3 Dahua DHI-ASC1202B-S

- Dahua DHI-ASC1202B-S – мережевий контролер для двох точок доступу [6]. Підтримує роботу з кодовими клавіатурами, зчитувачами карт, брелоків та біометричних даних. Дозволяє використовувати

комбінацію карток з паролями та відбитками пальців. Має 6 входів та 4 виходи. Для підключення до мережі використовує Ethernet інтерфейс.

Контролер керує доступом на дві односторонні або одну двосторонню точки проходу. Дозволяє зберігати 100000 кодів діючих карт та 150000 записів подій. Обладнаний двома інтерфейсами Wiegand та RS485 для підключення зчитувачів брелоків, карт та біометрії, кодових клавіатур. Підтримує використання кількох карток, а також поєднання карток, паролів та відбитків пальців.

Контролер Dahua DHI-ASC1202B-S оснащений шістьма вхідними роз'ємами: для двох кнопок, двох дверних датчиків та двох тривожних детекторів. Має 4 вихідні інтерфейси – 2 реле закриття дверей, 2 тривожні. Реалізовано функції сповіщення про тривожні події: перевищення часу очікування відкриття, спрацьовування охоронної сигналізації, вимушеної сигналізації, несанкціонованого проходу.
Пристрій дозволяє встановлювати заборону повторного проходу та здійснювати мультиблокування дверей. Є можливість розділити користувачів на групи постійних, VIP, гостей та тих, хто перебуває в чорному списку. Підтримується 128 груп із розкладом, 128 із періодами
Ціна: 96 доларів США

Рис. 4 Partizan PAC-12.NET

- Мережевий контролер доступу Partizan PAC-12.NET [7]

  Контролер на 1 двосторонню точку проходу.

  Кількість користувачів 30000, подій 60000.

  Підключення двох зчитувачів з інтерфейсом Wiegand 26/Wiegand 34, 1 кнопка виходу, 1 датчик стану дверей.

  Пожежне розблокування, енергонезалежна пам'ять, мережний інтерфейс TCP/IP, Живлення 12В/80мА, без джерела живлення.

  Ціна: 154 долари США

Рис. 5 U-Prox IC A

- Мережевий контролер доступу U-Prox IC A – це контролер, який призначений для організації систем контролю доступу на великих підприємствах та житлових об'єктах [8]. Головним завданням даного контролера є глобальне відстеження та захист від дублювання проходження по одній картці (ідентифікатору). За допомогою IC A можна поєднувати контролери доступу в одну систему, яка відстежуватиме переміщення співробітників по території. Контролер дозволяє відстежити повторний прохід співробітниками, використання або передачу між працівниками дублікатів карток, несподівану появу всередині території, передачу ідентифікатора іншим особам.

  Особливості пристрою:

  Можливість поділити приміщення на ділянки;

  Вхід та вихід через кілька точок;

  В однієї людини може бути кілька ідентифікаторів;

  Відстеження передачі ідентифікаторів.

Ціна: 192 долари США

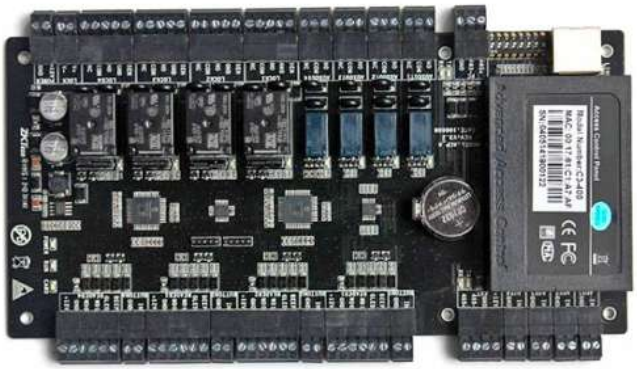

Рис. 6 ZKTeco C3-400

- Мережевий контролер доступу ZKTeco C3-400 для 4 дверей використовується для забезпечення контролю доступу в приміщення, що охороняються [9]. До нього можна підключати різні зчитувачі, ключем для яких є карти, брелоки та кодові клавіатури. З його допомогою можна створити систему підрахунку часу роботи для кабінету чи номера у готелі. Використовується для чотирьох дверей.

  Особливості:

  - Вбудована операційна система LINUX.
  - Пам'ять 30 000 RFID карт та 100 000 записів журналу подій.
  - Підтримка кількох форматів карт Wiegand та клавіатурного пароля.
  - Використання технологій зв'язку: Ethernet та шина RS485.
  - Вбудований в панель управління апаратний сторожовий таймер запобігає збоям.
  - Захист від перевантаження по струму, перенапруги та зворотної напруги для адаптера живлення, що підключається.
  - Захист від перевантаження струму для живлення зовнішніх зчитувачів.
  - Захист від перенапруги для всіх портів введення/виведення.
  - Захист від перенапруг для портів зв'язку.

  Ціна: 222 долари США

Рис. 7 Hikvision DS-K2802

- Hikvision DS-K2802 – надійний контролер доступу, який призначений для 2-х дверей [10]. Як і інші контролери серії, він відрізняється функціональністю та використанням самоадаптивного мережного інтерфейсу при передачі даних.

  Hikvision DS-K2802 - контролер доступу, який відрізняється високою надійністю та функціональністю. Він дозволяє обробляти сигнал шифруванням та підтримує автономну роботу. Пристрій вбудований у металевий корпус, який має компактні розміри та лаконічний дизайн.

  Головне призначення – ефективний контроль доступу до об'єкта у двох пунктах проходження. Встановлюється пристрій відносної близькості від дверей (або в іншому зручному місці) і стане відмінним рішенням для компанії або фабрики. У пам'яті всього зберігається 10 тис. карток та 50 тис. Подій.

  Ціна: 161 долар США

Інша частина важлива частина системи контролю доступу — зчитувач. Зазвичай вони мають універсальний вихідний роз'єм, тому сумісні з усіма вищепереліченими контролерами. Розглянемо варіанти зчитувачів, які доступні на ринку:

Рис. 8 Dahua DH-ASR1100A

- Dahua DH-ASR1100A – RFID зчитувач для карт Mifare [11]
  Особливості: протокол: RS-485, тип карт: IC (Mifare) card, синє підсвічування, функція Watchdog, звуковий сигнал та індикатор.
  Ціна: 39 доларів США

Рис. 9 Sparta SR10D

- Sparta SR10D призначений для системи контролю доступу для швидкого считивания карток або брелків стандарту EM Marine 125 Кгц, WG26 Потребує 12 - 14 В [12]. Зчитувач виготовлений в мініатюрному пластмасовому корпусі з подальшою його герметизацією. Завдяки малим габаритним розмірам, зчитувач володіє підвищеною міцністю і надійністю.

  Захист зчитувача від негативних впливів навколишнього середовища забепечується за рахунок заливки його плати компаундом.

  Екранований кабель для підключення зчитувача виведений з його тильної сторони.

  Є звукова/світлова індикація, також є сигнал зумера і двоколірний світлодіод.

  Ціна: 14 доларів США

Рис. 10 ATIS PR-01 MF

- Зчитувач ATIS PR-01 MF [13]

  Мініатюрний зчитувач брелоків стандарту Mifare, врізний монтаж.

  Зчитувач ATIS PR-01 MF застосовується в системах контролю і керування доступом для передачі контролеру коду піднесеного до зчитувача брелока стандарту Mifare через протоколи Wiegand 26.

  Технічні характеристики:

  - Робоча частота: 13,56 МГц
  - Зчитування ідентификаторів (брелоки, браслети): Mifare Ultralight, Mifare Standard (Classic) 1K і 4K, Mifare ID
  - Дальність зчитування: 1-4 см
  - Вихідний інтерфейс: Wiegand 26
  - Світлова індикація: є
  - Напруга живлення: 8-15В постійного струму
  - Споживання струму: 35мА
  - Габаритні розміри (мм): D26XH22

  Ціна: 20 доларів США

Рис. 11 ARNY AR-M10 EM

- RFID зчитувач карток EM-Marine U-Prox mini 485 [14] призначений для використання в системах контролю та управління доступом. Він використовується для підключення через інтерфейс RS-485 та роботи з ідентифікаторами ASK, FSK або ASK+FSK.

  Зчитувач виконаний у міцному водонепроникному пластиковому корпусі та має ступінь захисту IP66, завдяки чому може використовуватись для встановлення у вуличних умовах. Ось ще деякі характеристики цього рідера:

  тип карток: EM-Marine;

  дальність зчитування безконтактних карток: до 8-10 см;

  інтерфейс підключення: RS-485;

  наявність світлодіодної (червоний та зелений) індикації та зумера;

  Ціна: 37 доларів США

З розглянутих систем робимо висновок, що система складається з контролера, засобу ідентифікації та хабу в якості якого використовується персональний комп'ютер. Система має певну базу ідентифікаторів, для яких дозволено виконання поставлених задач, таких як відкриття замка, вмикання двигуна автомобіля, запис в електронний графік робочого часу та ін.

В залежності від складності системи, обирається контролер, або може бути взагалі відсутнім, використовуючи на пряму зав'язок з хабом.

Ідентифікатором користувача може бути ID карта, QR код, відбиток пальця, радіо-маячок, пароль введений через клавіатуру .

Недоліки оглянутих систем:

- висока ціна
- використання стандартизованого програмного наданого розробником, що підвищує можливість злому, та унеможливлює користувацьку модернізацію системи
- відсутність єдиного автономного хабу для деяких систем
- великі габарити контролерів системи

2. ОБҐРУНТУВАННЯ ТЕХНІЧНИХ РІШЕНЬ

В цьому розділі розглядається загальна структура системи, обираються технології які найкраще підходять для інтеграції систему безключового доступу.

2.1 Структурна схема

На рисунку 12 показано структурну схему головного модуля.

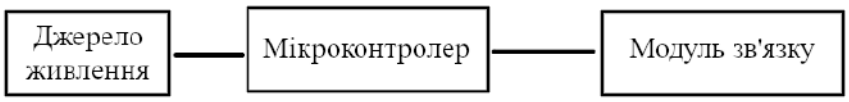

Рис. 12 головний модуль системи

Головний модуль системи використовується в якості хабу. Він виконує запити на сервер, головний комп'ютер, чи використовує бібліотеку дозволених ідентифікаторів які знаходяться в портативному носії даних, наприклад модуль для портативної карти пам'яті.

Його завдання отримати рад ідентифікаторів та передати їх на відповідний виконавчий модуль, використовуючи модуль зв’язку.

На рисунку 13 показано структурну схему виконавчого модуля.

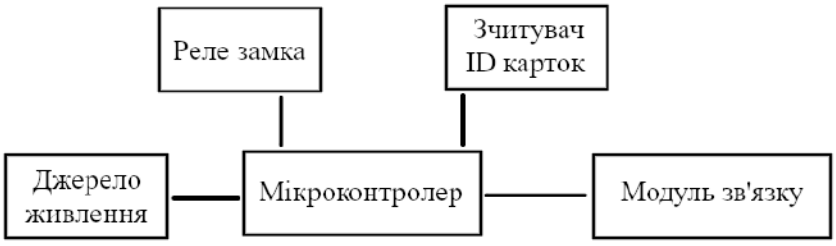

Рис. 13 виконавчий модуль

Виконавчий модуль — пристрій який визначає, чи пропустити власника ідентифікатора у двері, оскільки зберігає коди ідентифікаторів зі списком

прав доступу кожного з них у власній пам'яті. Для цього виконується з’єднання з головним модулем, оновлюється база ідентифікаторів. Коли ключа карта підноситься до зчитувача, контролер порівнює предявлений ключ з наявною базою ключів, після чого приймається рішення про відкриття дверей за допомогою реле.

2.2 Вибір контролера

Для зібрання пристрою необхідно одрати мікроконтролер, який буде використано для проектування пристрою.

2.2.1 Raspberry Pi PICO

Raspberry Pi PICO - це плата розробника від компанії Raspberry Pi Foundation основою якої є двоядерний мікроконтролер (власна розробка компанії) ARM Cortex-M0, максимальна тактова частота якого становить 240 МГц. Також на платі встановлено флеш-пам'ять об'ємом 2Мб. У разі потреби обсяг флеш-пам'яті можна розширити до 16Мб за допомогою шини QSPI. Об'єм вбудованої оперативної пам'яті складає 256Кб [15].

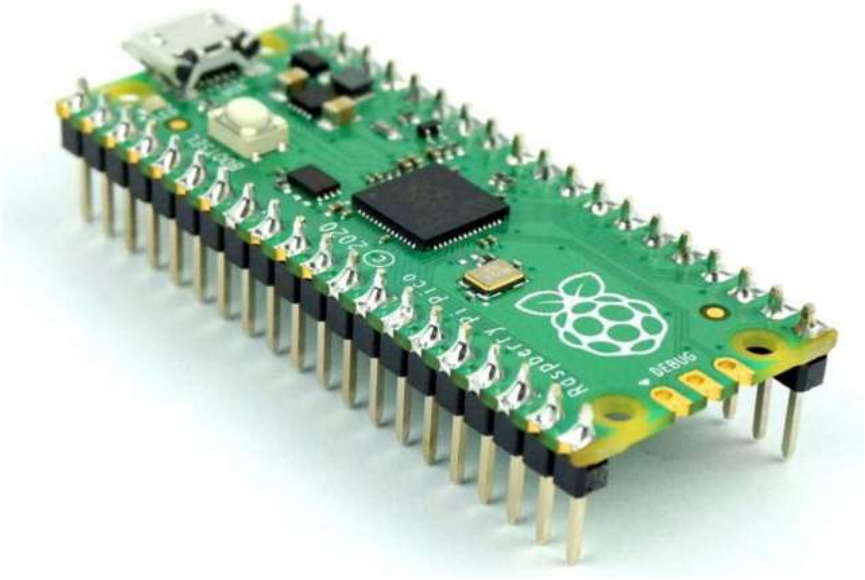

Рис. 14 Raspberry Pi PICO

Користувачеві надається 26 багатофункціональних контактів GPIO (контакти по обидва боки плати з кроком 2.54 мм). Доступні інтерфейси: 2×SPI, 2×I2C, 2×UART, 3×12-бітних АЦП, 16 керованих каналів ШІМ. Реалізовано режим сну з низьким енергоспоживанням, 8 програмованих кінцевих автоматів вводу-виводу (PIO) для периферійних пристроїв, вбудовані бібліотеки для обчислень з плаваючою точкою, вбудований датчик температури, один світлодіод користувача (зелений) і роз'єм налагодження (контакти з протилежного боку плати від роз'єму microUSB крок 2.54 мм). Програмування та живлення плати можливе через порт microUSB. При використанні плати у кінцевому пристрої живлення можна реалізувати зовнішнє, без використання роз'єму microUSB. Напруга живлення може бути від 1.8 В до 5.5 В.

2.2.2 Arduino UNO

Репліка оригінальної плати Arduino UNO [16]. Як USB-UART перехідник використовується мікросхема CH340, яка добре себе зарекомендувала та відрізняється хорошою стабільністю, високою швидкістю передачі даних, але для якої потрібна додаткова установка драйверів.Також контролер від попередників відрізняється додатковими контактами SDA і SCL (I2C інтерфейс) і виходами AREF - джерела опорногї напруги для АЦП контролера і IOREF - виходом напруги живлення портів введення-виведення (для автоматичного перемикання напруги периферії при використанні 5В і 3.3В контролерів). В усьому іншому це все той же контролер Arduino UNO на базі мікроконтролера Atmega328p з масою прикладів програм, бібліотек і опису побудови готових конструкцій.

Рис. 15 Arduino UNO

Характеристики:

Мікроконтролер : ATmega328

Цифрові входи/виходи: 14 (6 з них ШІМ)

Аналогові входи: 6

Флеш-пам'ять програм: 32Кб

Оперативна пам'ять: 2Кб

Тактова частота: 16 МГц

Розмір: 68 х 53 х 15 мм

2.2.3 NodeMCU

NodeMCU – це недорога платформа IoT з відкритим кодом.

Спочатку він включав мікропрограму, яка працює на ESP8266 Wi-Fi SoC від Espressif Systems, і апаратне забезпечення з підтримкою 32-розрядного

MCU ESP32. Наявність вбудованого Wifi на платі аргументоване можливістю реалізувати доступ до інтернету через найближчу точку доступу[17]. Це дозволить завантажувати пакети з певного ресурсу для оновлення даних на платі. Цей модуль оснащений вбудованим USB-роз'ємом і багатим набором розпинівки.

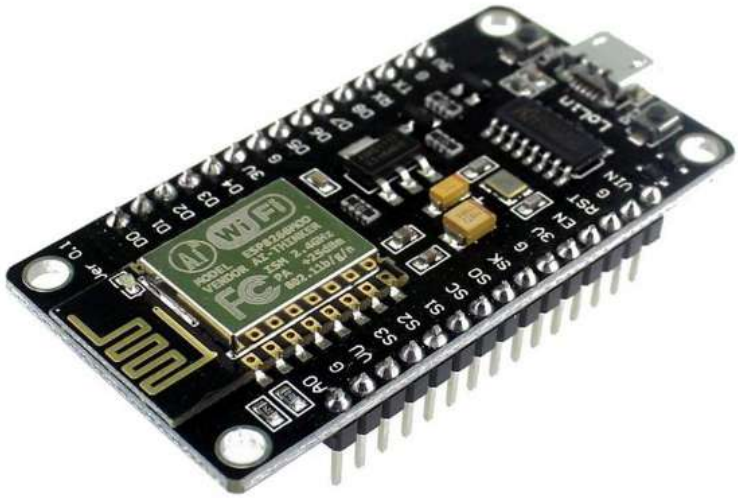

Рис. 16 NodeMCU v3

Характеристики:

• Напруга: 3,3 В.

• Wi-Fi Direct (P2P), soft-AP.

• Споживання струму: 10uA~170mA.

• Флеш-пам'ять, що підключається: максимум 16 МБ (звичайно 512 КБ).

• Інтегрований стек протоколів TCP/IP.

• Процесор: Tensilica L106 32-розрядний.

• Частота процесора: 80~160 МГц.

• ОЗП: 32K + 80K.

• Аналогово-цифровий: 1 вхід з роздільною здатністю 1024 кроки.

• Підтримка 802.11: b/g/n.

• Максимальна кількість одночасних TCP-з'єднань: 5.

2.2.4 плата FONA

Універсальна плата розробника FONA на мікроконтролері Atmega32U4, що працює при напрузі живлення 3.3В на частоті 8МГц від компанії Adafruit. Поєднує в собі всю міць AVR мікроконтролера з універсальним радіо-модулем SIM800. Плата ідеально підходить для автономних пристроїв від-даленого збору, обробки та передачі інформації з датчиків. На платі є все необхідне для підключення LiPO акумулятора та має вбудований зарядний пристрій, що живиться від microUSB роз'єму [19].

Рис. 17 плата FONA

Характеристики:

Розміри: 61 x 23 x 7 мм

Мікроконтролер: ATmega32u4 @ 8MHz

Пам'ять для програм: 32К

Пам'ять даних: 2К

Радіомодуль: SIM800

Напруга живлення: 3.3В

Максимальний споживаний струм: 500мА

Завантаження програм: через USB порт

Цифрові входи/виходи: 20

Інтерфейси: UART, I2C, SPI

ШІМ: 8

Аналогові входи: 10

Вбудований стабілізатор напруги: 3.3В 500мА

Світлодіод: на 13 піні

2.2.5 обрана плата

Для роботи було обрано мікроконтролер NodeMCU v3, так як він має необхідну кількість виводів, має інтерфейс SPI, а також має вбудований Wifi модуль, який буде використовуватись головним модулем для подальшого впровадження з’єднання з сервером, через який буде виконуватись оновлення ключів доступу для виконавчих модулів.

2.3 Передавання даних

Виходячи з умов завдання, необхідно об’єднати контролери в мережу керованих пристроїв. Для цього в конструкції присутній модуль зв’язку, який виконує передачу даних між елементами системи.

Передавання даних — процес передачі цифрового (бітового) потоку від джерела до певної точки або множити точок, для подальшого оброблення комп'ютерними засобами [20]. Зазвичай для передачі даних використовуються мідні проводи, оптичне волокно, бездротові канали зв'язку.

Сигнали передаються в цифровій та аналоговій формах [21].

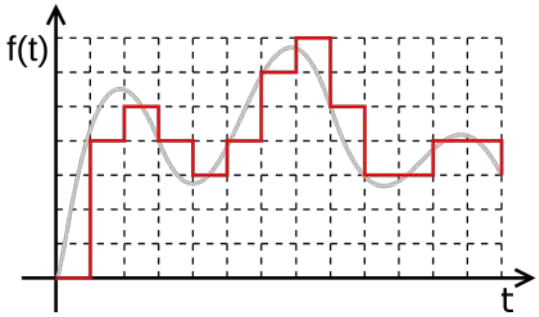

Рис. 18 Оцифрований аналоговий сигнал

Переваги використання цифрових сигналів:

- більша захищеність від дії шумів, перешкод;
- невеликі відхилення від дозволених значень ніяк не викривляє цифровий сигнал, так як завжди існують зони допустимих відхилень;
- дозволяє складнішу і багатоступеневу обробку
- більш довге зберігання без втрат
- якісніша передача
- цифрові пристрої легше проектувати, відлагоджувати.

На ринку існують різні технології, які можуть бути використані для даного проекту. Розглянемо деякі з них [22].

2.3.1 Дротові технології

Використання дротів — довлі просте в застосуванні. Для підключення дротів необхідно мати відповідні інтерфейси у пристрої. Обмін даними між платами Arduino можна реалізувати за допомогою послідовного інтерфейсу. Для цього використовуються контакти RX та TX. При використанні серійного інтерфейсу дані передаються в байтах. Ці байти зчитуються другим Arduino по одному.

Даний варіант є доволі простим у виконанні, але не підходить для даного проекту. Недоліками вказаного варіанту є:

- так як необхідно використати велику кількість дроту, що є доволі затратним.
- для встановлення дротів у приміщеннях необхідно проводити додаткові монтажні роботи.

2.3.2 Інфрачервоне випромінювання

Усі інфрачервоні бездротові мережі використовують для передачі даних інфрачервоні промені. У подібних системах необхідно генерувати дуже сильний сигнал, оскільки інакше значний вплив робитимуть інші джерела, наприклад світло з вікна. Цей спосіб дозволяє передавати сигнали з великою швидкістю, оскільки інфрачервоне світло має широкий діапазон частот. Інфрачервоні мережі здатні нормально функціонувати на швидкості 10 Мбіт/с.

Для використання в Arduino необхідно використовувати пару приймач-передавач, наприклад: KY-005 і KY-022.

Характеристики:

- Несуча частота: 38 кГц
- Дистанція прийому від звичайного пульта: 18-20 м
- Кут прийому: 90 градусів
- Напруга живлення постійного струму: 2,7-5,5 В

Дана технологія вимагає знаходження приладів в зоні прямої видимості, але вже має переваги у вартості та компактності у порівнянні зі звичайним дротом.

2.3.3 Лазерні технології

Лазер — джерело когерентного, монохроматичного і вузькоспрямованого електромагнітного випромінювання оптичного діапазону, яке характеризується великою густиною енергії. Існують газові лазери, рідинні та на твердих тілах.

Для arduino достатньо використати фоторезистор VT90N в якості приймача та KY-008 в якості передатчика.

Лазер має високі швидкісні показники, технологія схожа на інфрачервону тим, що вимагає прямої видимості між передавачем і приймачем. Якщо з якої-небудь причини промінь буде перерваний, то це перерве і передачу. Це робить нераціональним використання даної технології в великих закритих приміщеннях.

2.3.4 Радіо Ethernet (IEEE 802.11)

IEEE 802.11 — це родина технологій бездротового передавання в радіодіапазоні. Сьогодні найпопулярніша технологія стандарту IEEE 802.11b; вона дає змогу передавати дані зі швидкістю 11 Мбіт/с на відстань від кількох до десятків кілометрів. Вихідна швидкість залежить від рівня завад, обладнання. На базі IEEE 802.11b будують безпроводові локальні мережі Wireless LAN.

На arduino можливо розмістити модуль ESP8266 або взяти плату з вбудованим модулем.

Технологія використовує високі частоти, що практично унеможливлює передавання сигналу через велику кількість стін у будівлях. Для компенсації цього ефекту можливо використати систему підстанцій, тобто головний передавач передає сигнал на підстанцію, яка передає його далі або до виконавчого пристрою.

2.3.5 Bluetooth

Bluetooth — це інтерфейсна безпроводова технологія. Діаметр мережі 10-30 м (у перспективі — 100 м). Працює в багатопунктовому режимі, не

обов'язково в зоні прямої видимості. Головне призначення — створення побутових мереж, приєднання мультимедійної периферії.

Щоб використовувати цю технологію достатньо встановити Bluetooth модуль HC-06.

Bluetooth має малу зону покриття, що робить його менш придатним для використання на великих відстанях.

2.3.6 LoRa

LoRa – технологія модуляції малопотужної мережі передачі даних зі швидкістю 0,3-50 кб/с та дальністю від 1-2 до 10-15 км (в ідеальних умовах). в діапазоні частот, що не ліцензується. Використовується для передачі даних в автономних датчиках екологічного спостереження та комунальному господарстві [23].

2.3.7 GSM

GSM. Перевага використання бездротового каналу зв'язку - практично суцільне покриття. До основних методів передачі інформації в мережі, що розглядається, відносяться GPRS, SMS і голосовий канал.

Вказана технологія не буде використовуватись в даному проекті, так як вимагає використання мережі базових станцій.

2.3.8 Zigbee

Zigbee — стандарт для набору високорівневих протоколів зв'язку, що використовують невеликі малопотужні цифрові трансівери, заснований на стандарті IEEE 802.15.4-2006 для бездротових персональних мереж, таких як, наприклад, бездротові навушники, з'єднані з мобільними телефонами за допомогою радіохвиль короткохвильового діапазону[24]. Технологія ви-

значається специфікацією Zigbee, розробленою з наміром бути простіше та дешевше, ніж інші персональні мережі, такі як Bluetooth. Zigbee призначений для радіочастотних пристроїв, де необхідна тривала робота від батарейок та безпека передачі даних через мережу.

Основними областями застосування технології Zigbee є бездротові сенсорні мережі, автоматизація житла («Розумний дім» та «Інтелектуальна будівля»), медичне обладнання, системи промислового моніторингу та управління, а також побутова електроніка та «периферія» персональних комп'ютерів.

2.3.9 Модуль передавання даних

Для подальшого використання обрано приймач SX1278. Він оснащений модемом дальнього радіуса дії LoRa , який забезпечує над широкосмуговий зв'язок з розширеним спектром і високу стійкість при мінімальному споживанні струму [25]. Пристрій має високу чутливість, що робить його оптимальним варіантом для використання на об'єктах інфраструктури.

Рис. 19 LoRa 433 мГц

Таблиця 2.2 — Параметри модуля LoRa SX1278 [26]

Параметр	Значення
Метод модуляції:	FSK/GFSK, LoRa
Тип зв'язку:	напівдуплексних зв'язок
Робоча частота:	137 - 525 МГц
Фактор поширення:	6 - 12
Ширина каналу:	7.8 - 500 кГц
Ефективний бітрейт:	0.18 - 37.5 kbps
Встановлена чутливість:	-111 to -148 dBm
Відхилення каналу (ADJ):	56 dBm
Робочий діапазон:	ISM multiband
Енергоспоживання в режимі прийому:	12 ~ 13 мА
Внутрішній буфер:	256 байт FIFO TX/RX
Виявлення сигналу каналу передачі даних:	ISSI

2.4. Вибір зчитувача

Зчитувач використовується для зняття інформації про доступ користувача за допомогою ідентифікатора. На ринку існує декілька технологій, які сумісні з обраним контролером.

2.4.1 Сканер штрих-кодів

Сканер штрих-кодів від Waveshare являє собою мініатюрний сканер 1D / 2D-кодів, який здатний за допомогою внутрішнього інтелектуального алгоритму розпізнавання зображень швидко і точно декодувати штрих-код або QR-

код з паперу або екрану будь-якого пристрою. Завдяки вбудованому USB / UART інтерфейсу, сканер безпосередньо підключається до комп'ютера або іншого мобільного пристрою[27].

Рис. 20 Barcode Scanner Module, 1D / 2D Codes Reader

Переваги використання:

- Використання штрих-кодів, QR-кодів для доступу
- наявність вмонтованого зумера(звук при зчитуванні)

Недоліки:

- ціна продукту
- масивність модуля

2.4.2 Модуль сканера відбитків пальців

Ємнісний напівпровідниковий модуль сканера відбитків пальців з вбудованим процесором STM32F105, в якому використаний комерційний алгоритм розпізнавання відбитків пальців. Дозволяє швидко та стабільно перевіряти відбитки пальців [28].

Виконано на основі високошвидкісного цифрового мікроконтролера STM32F105R8 в поєднанні з високопродуктивним комерційним алгоритмом відбитків пальців і вдосконаленим напівпровідникових датчиком, ємнісний пристрій для читання відбитків пальців стає простим, але інтелектуальним модулем інтеграції, забезпечує такі функції, як реєстрація відбитків пальців, обробка зображень, пошук функцій, створення та зберігання шаблонів, зіставлення та пошук відбитків пальців і т. д.

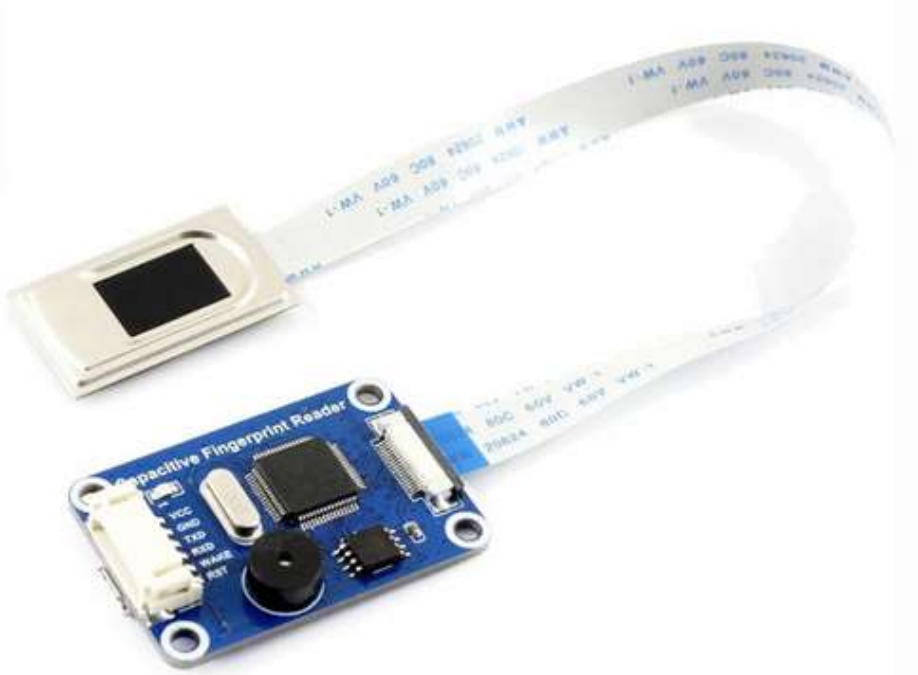

Рис. 21 Barcode Scanner Module, 1D / 2D Codes Reader

Переваги:

- не потребує закупки додаткових предметів для ідентифікації (картки, штрих-коди)
- компактний модуль

Недоліки:

- ціна
- масивність простору зайнятого великою кількість зображеннями відбитків

2.4.3 Зчитувач ID карток

RFID - спосіб автоматичної ідентифікації об'єктів, у якому за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в так званих транспондерах, або RFID-мітках[29].

Будь-яка RFID-система складається з зчитувального пристрою (зчитувач, рідер або Інтеррогатор) і транспондера (він же RFID-мітка, іноді також застосовується термін RFID-тег).

Більшість RFID-міток складається з двох частин. Перша - інтегральна схема для зберігання та обробки інформації, модулювання та демодулювання радіочастотного сигналу та деяких інших функцій. Друга - антена для прийому та передачі сигналу.

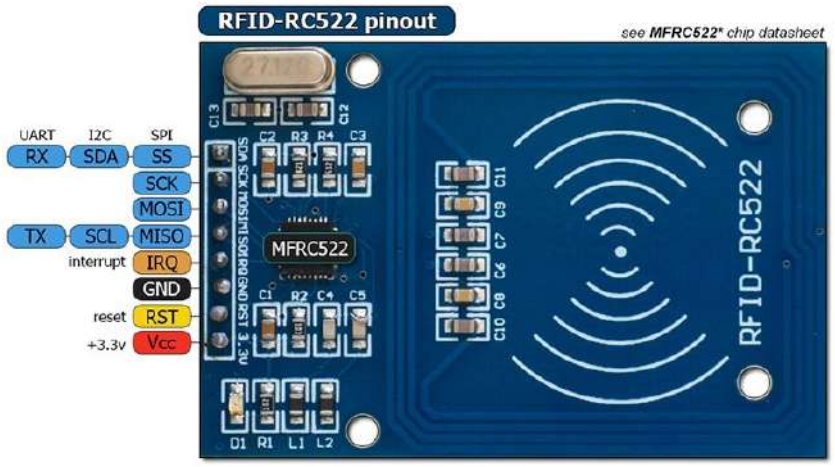

Рис. 22 RFID-RC522

Основні характеристики [30]:

Заснований на мікросхемі MFRC522

Напруга живлення: 3.3V

Струм: 13-26mA

У режимі очікування: 10-13ma

У сплячому режимі: менше 80 мкА

Робоча частота: 13.56MHz

Дальність зчитування: 0 ~ 60 мм

Інтерфейс: SPI, максимальна швидкість передачі 10Мбіт / с

Розмір: 40мм х 60мм

Переваги:

- компактне недороге рішення
- простота у використанні

Недоліки:

- можливість копіювання ідентифікатора

Для проекту обрано технологію RFID міток, так як вона потребує менеш місця для ідентифікаторів у порівнянні зі сканером відбитків пальців. RFID для копіювання потребує безпосереднього зчитування, в той час як QR код достатньо сфотографувати. Додатковою перевагою використання технології є простота заміни ключа та загальна ціна рішення.

2.5 Джерело живлення

Так як використаний контролер потребує живлення 1 ампер 5 вольт, та установка є стаціонарною, використаємо звичайний блок живлення.

Найпростіший варіант — використати живлення від мережі за допомогою зарядного пристрою який та підключеного до нього microUSB, так як саме такий роз'єм використовується обраним мікроконтролером. Так як конструкція доволі проста, для впровадження безперебійного живлення доведеться додати звичайний павербанк до джерела живлення.

Рис. 23 зарядний пристрій з USB-роз'ємом

Малопотужний мініатюрний зарядний пристрій з USB-роз'ємом. Вихідна напруга 5 В, струм до 500 мА. Колір корпусу чорний або білий.

5В 1000мА 2xUSB Type A

Характеристики:

Вхідна напруга: 100-240 В

Частота: 50/60 Гц

Вих. напруга: 5 В

Вих. сила струму: до 500 мА

Вихідний роз'єм: USB-A Female

Рис. 24 USB 2.0 кабель microUSB-USB

3. МАКЕТУВАННЯ СИСТЕМИ БЕЗКЛЮЧОВОГО ДОСТУПУ

В цьому розділі проводиться зібрання макету з використанням обраних варіантів технічних рішень, написання коду для пристрою та його запуск.

3.1 Апаратна частина пристрою

Головним елементом макету є мікроконтролер, до якого будуть кріпитися всі ніші деталі збірки. Почнемо збірку макету з приєднання радіо. Використаний радіомодуль використовує декілька вільних цифрових пінів та шину SPI [18] [35].

SPI — фактичний послідовний синхронний повнодуплексний стандарт передачі даних, розроблений фірмою Motorola для забезпечення простого сполучення мікроконтролерів та периферії. SPI також називають чотирьохпровідним (англ. four-wire) інтерфейсом.

Для передачі даних в інтерфейсі SPI використовуються чотири сигнали:

MOSI або SI — вихід ведучого, вхід веденого. Служить для передачі даних від ведучого пристрою до веденого.

MISO або SO — вхід ведучого, вихід веденого. Служить для передачі даних від веденого пристрою до ведучого.

SCLK або SCK — послідовний тактовий сигнал. Служить для передачі тактового сигналу для ведених пристроїв.

CS або SS — сигнал початку/завершення сеансу зв'язку. По завершенні обміну даних має бути знятий, що дасть змогу приймачу даних вийти з режиму читання/запису та перейти до режиму обробки даних.

Особливість інтерфейсу при підключенні: MOSI, MISO, SCLK — визначені як конкретні піни на платі, а CS — довільний цифровий пін.

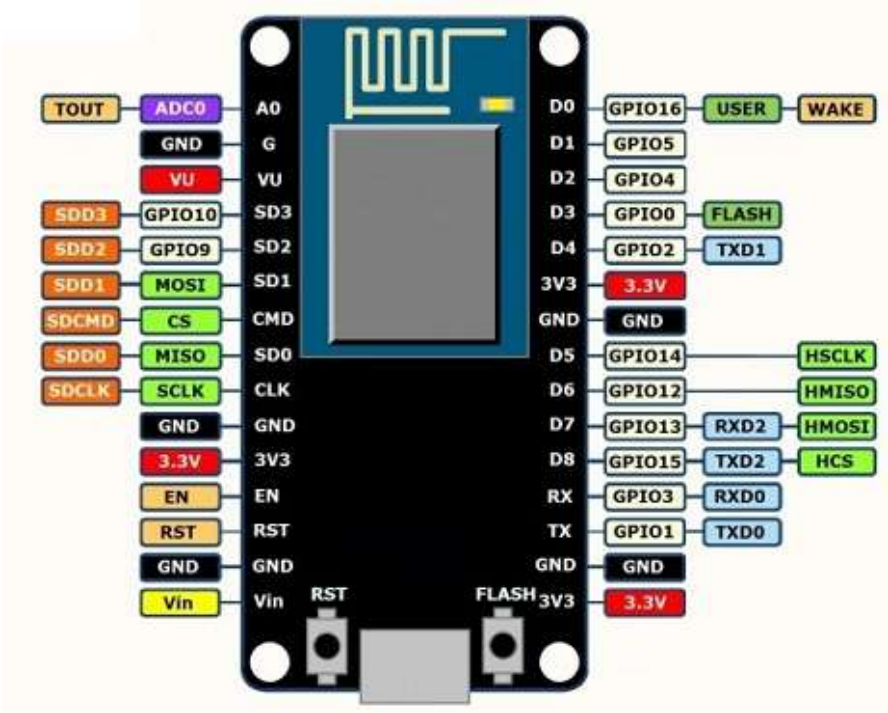

Рис. 25 піни плати NodeMCU-V3

Скорочення H перед назвою піну SPI означає (рис. 25), що цей вивід належить до другої шини SPI на даній платі.

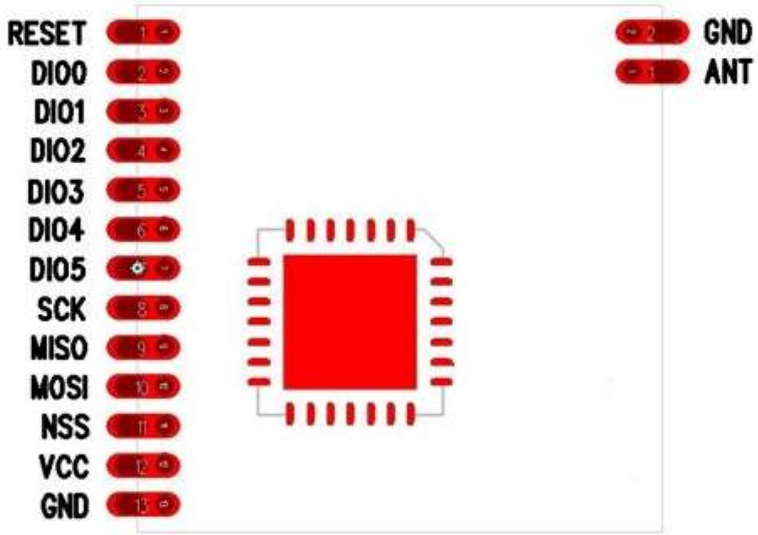

Рис. 26 піни радіомодуля LoRa

Для з'єднання модулів використаємо наступну схему підключення:

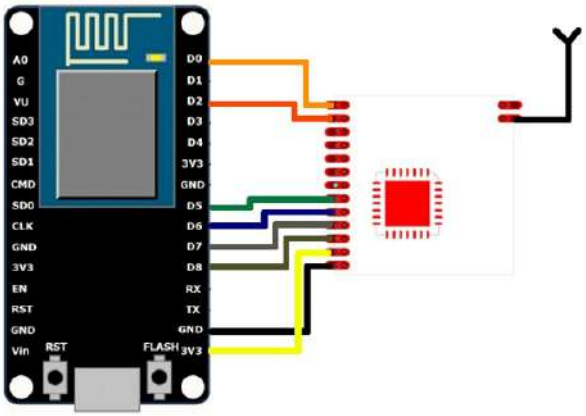

Рис. 27 зібрана схема передатчика

Для передачі даних між платами використаємо асинхронну схему передачі, тобто сигнал синхронізації між двома елементами буде відсутній. Тож для якісної передачі інформації використовуються пакети даних, де перший пристрій передає пакет, а інший очікує на код початку пакету.

Інформація в середині пакету передається символами з відомого кодового набору, такого як ASCII. Тобто в середині пакету використовується «посимвольне пересилання»

До виконавчого пристрою також під'єднуємо радіомодуль як вказано на рис. 27. Модуль RC-522 також підключається на шину SPI, використовуючи рис. 22. Для відмикання дверей буде використано реле на яке подається сигнал. Реле має чотири виводи, два з яких призначені на підключення керуючого ланцюга, а 3-тій та 4-тий вмикаються в коло під навантаження, в даному випадку в ланцюг замка. При подачі керуючого сигналу на реле, основне коло замикається.

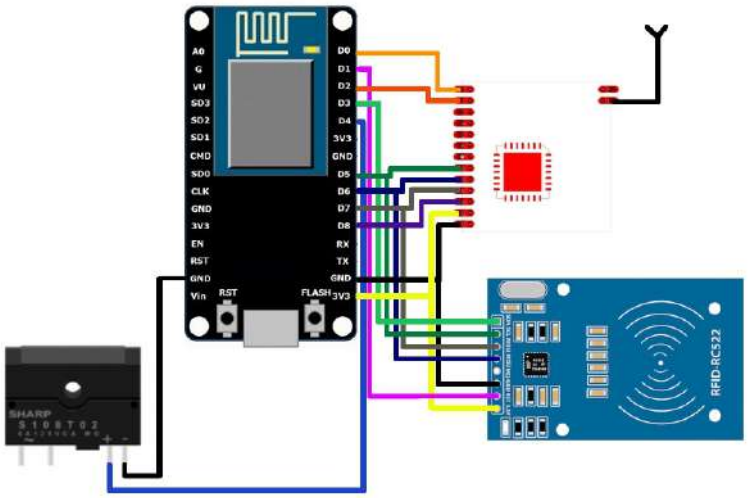

Рис. 28 зібрана схема виконавчого пристрою

Система має два зібраних пристрої, при чому виконавчий пристрій має обирати пристрій по шині SPI який буде працювати в конкретний момент часу.

4.2 Програмна частина пристрою

Для написання коду було використано програму Arduino IDE [33].

Програмне забезпечення Arduino з відкритим кодом дозволяє легко писати код і завантажувати його на плату. Це програмне забезпечення можна використовувати з будь-якою платою Arduino.

Для написання програмної частин приладу використано мову програмування C [34].

C - компілювана, статично типізована мова програмування загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування.

C — мінімалістична мова програмування. Серед її головних цілей: можливість прямолінійної реалізації компіляції, використовуючи відносно простий компілятор, забезпечити низькорівневий доступ до оперативної пам'яті, формувати лише кілька інструкцій машинної мови для кожного елементу мови і не вимагати великої динамічної підтримки. У результаті код С придатний для більшості системного програмного забезпечення, яке традиційно писали асемблером.

Незважаючи на її низькорівневі можливості, мову проєктували для платформонезалежного програмування. Сумісна зі стандартами та платформонезалежна написана мовою С програма може легко компілюватися на великій кількості апаратних платформ та операційних систем з мінімальними змінами. Мова стала доступною для великої кількості платформ - від вбудованих мікроконтролерів до суперкомп'ютерів.

Основні особливості Сі:

проста мовна база, з якої до стандартної бібліотеки винесено багато суттєвих можливостей, на кшталт математичних функцій або функцій роботи з файлами;

- орієнтація на процедурне програмування;
- система типів, що оберігає від безглуздих операцій;
- використання препроцесора для абстрагування однотипних операцій;
- доступ до пам'яті через використання покажчиків;
- невелика кількість ключових слів;
- передача параметрів у функцію за значенням, а чи не за посиланням (передача за посиланням емулюється за допомогою покажчиків);

- наявність покажчиків на функції та статичні змінні;
- області видимості імен;
- структури та об'єднання - визначальні користувачем збірні типи даних, якими можна маніпулювати як одним цілим.

Частина коду призначена для головного модулю:

```
#include <SPI.h> //підключення бібліотеки для керування SPI переиферією
#include <LoRa.h> //підключення бібліотеки радіомодулем  LoRa SX1278
#include <Arduino_JSON.h> //підключення бібліотеки для JSON коду
```

На початку програми задаємо перелік бібліотек, які будуть використовуватися компілятором для інтерпритації коду.

Задаємо значення пінів для використання модулю LoRa, та деякі інші змінні для роботи програми:

```
#define ss 15  //задаємо назву ss для піна 15
#define rst 16 //задаємо назву rst для піна 16
#define dio0 2//задаємо назву dio для піна 12


int counter = 0;  //оголошуємо ряд змінних для виконання програми
String jsonString;
String jsonString2;

int i, n;
JSONVar JSArray;

//змінні для керування ключами
 byte DataKey1[]    = {0, 0, 0, 0};
int key1size = 4;

char key[26];
```

```
void setup() {
  Serial.begin(9600);
  while (!Serial); // очікування COM порта, закоментувати, якщо не буде підєднано
  Serial.println("LoRa Sender");

  LoRa.setPins(ss, rst, dio0); // назначення виводів керування для радіомодулю

  JSONCreation(); // виклик функції для створення ключа

  if (!LoRa.begin(433E6)) {  // вхід в цикл при помилці підключення радіомодуля
    Serial.println("Starting LoRa failed!");
    while (1){
      Serial.print(".");
      delay(1000);
     };
  }
}
```

Дана частина коду виконується лише один раз, після запуску, в якості попереднього налаштування мікроконтролера.

Для прототипу використано функцію очікування відкриття серійного порту для обміну даними з комп'ютером через COM порт. Він використаний для можливості отримання звітності від контролера про виконання певних частин коду. Для готового продукту дану функцію можна вимкнути, або залишити як журналу роботи пристрою, який буде передаватися на сервер.

Радіомодуль LoRa SX1278 запущено на частоті 433 мГц, що належить ISM діапазону частот [36]. ISM діапазон - частина радіочастотного спектра загального призначення, яка може бути використана без ліцензування.

```
void loop() {
  Serial.print("JSON packet: ");
  Serial.println(counter);
```

```
  // відправка пакету
  LoRa.beginPacket();
  LoRa.print(jsonString2);
  LoRa.endPacket();
  counter++; // нумерація пакету
  delay(5000); // затримка для економії енергії
}
```

Вищенаписана частина коду виконується циклічно і є основною частиною програми. В ній ведеться передача готового повідомлення, яке записано в змінній jsonString2, за допомогою радіомодуля, а також виводиться звітність про кількість переданих повідомлень до серійного порту.

```
void JSONCreation()
{
  Serial.println("========");

DataKey1[0] = 0xFE; // запис байту даних до рядку
DataKey1[1] = 0x96;
DataKey1[2] = 0x0E;
DataKey1[3] = 0xA9;

for (byte j = 0; j < key1size; j++)
  {
    JSArray[j] = DataKey1[j];  // запис рядку байтів до рядку JSON
  }
  jsonString = JSON.stringify(JSArray); // створення строки
  Serial.println(jsonString);
JSONmix();  // виклик функції шифрування
}
```

Функція JSONCreation() використовується для створення ключа, який буде передаватися, а також використовує JSON.

JSON — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передавання структурованої інформації через мережу що і буде впроваджено в пристрій для обміну ключами з сервером.

```
void JSONmix() {

 String sedit = jsonString;  // копіювання рядку
 char key1[50];
 String key     = "8372140965KLMNOPQRSTUVWXYZ";   // шифрований відповідник
String keyprot = "0123456789KLMNOPQRSTUVWXYZ";  // маска ключа
//"ABCDEFGHIJKLMNOPQRSTUVWXYZ"

for (int i = 0, n = jsonString.length(); i < n; i++)
   {
     key1[i] = jsonString[i]; // запис значення в буфер
     //Serial.println(jsonString[i]);

       for (int j = 0; j < 26; j++)
       {
          if (key1[i] == keyprot[j]) // порівняння буферу з маскою ключа
          {
             sedit [i] = key[j];  // заміна символу відповідно до шифру
          }
       }
   }
  jsonString2 = sedit;
    Serial.println(jsonString2);  //виведення результату
}
```

Вищеописана функція використана для можливості шифрування ключа який передається. Тобто усі символи у повідомленні порівнюються зі змінними у рядку keyprot та замінюються на символи рядка key використовуючи

їхній номер розміщення у рядку. Наразі використовується лише циферна частина, а букви — резерв символів, який може бути розширений.

Код для виконавчого модуля, на початку якого так само задаються змінні та бібліотеки для їх подальшого використання:

```
#include <SPI.h> //підключення бібліотеки для керування SPI переиферією
#include <LoRa.h> //підключення бібліотеки радіомодулем  LoRa SX1278
#include <Arduino_JSON.h> //підключення бібліотеки для JSON коду
#include <MFRC522.h> //підключення бібліотеки для керування зчитувачем

// піни для RC522
#define RST_PIN      D1
#define SS_PIN       D3

// піни для  модуля Lora
#define ss 15
#define rst 16
#define dio0 2

#define Dor D4 //пін для відмикання (Door relay)

//змінні для програми
int i = 0;
int LenghtInput = 63;
int temp = 0;
int truekey = 0;
int dataload = 0;
int cycle = 0;

JSONVar JsObject;
JSONVar JsObject1;
String jsonString;
String jsonStringmix;
String jsonString3;
```

```
char input[63];

byte dataBlock[]    = {0, 0, 0, 0, 0, 0, 0, 0,};
 byte block;
 byte len;
 byte BlockSize;
 // створення значень для зняття вимірів із зчитувача
 MFRC522::StatusCode status;
 MFRC522::MIFARE_Key key;
 MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup() {
  Serial.begin(9600);
  //while (!Serial);


  LoRa.setPins(ss, rst, dio0);
// вхід в цикл при помилці підключення радіо
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1) {
      Serial.print(".");
      delay(1000);
    };
  }

  Serial.println("LoRa Receiver");
  pinMode(LED_BUILTIN, OUTPUT);

   //активація піну для керування дверима
   pinMode(Dor, OUTPUT);
   digitalWrite(Dor, LOW);
}
```

В частині налаштувань також ініціалізується серійний порт, радіомодуль та задається режим виводу на пін призначений для взаємодії з реле.

```
void loop() {
// перевірка наявності вхідного символу через компорт
  if(Serial.available())
  {serial_input();}
// ##########################################

   switch (dataload) {  //вибір відповідного кейсу в залежності від змінної
     case 0:    //використання радіо

    LoraUsed();

       break;
     case 1:     //використання зчитувача

     mfrc522.PCD_Init();
     MFRC522doing();
       break;
   }
}
```

Основна частина коду, яка виконується циклічно, на початку має перевірку на наявність змінної в серійному порті. Ця функція використовується для керування контролером за допомогою серійного порту. Інша частина використовує оператор switch, якийпорівнює значення змінної dataload із значенням, визначеним в операторах case, що використовується для зміни режимк роботи пристрою між режимами приймання радіоповідомлень та зчитування ID карток.

```
void LoraUsed(void) {
  //спроба розібрати пакет
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    //пакет отримано
    Serial.print("\nReceived packet: ");
```

```
    //читаємо пакет
    if (LoRa.available()) {
      LoraPacketType();
      JsonPacket();
    }
  }
}


void LoraPacketType(void) {
 i = 0;
  while (LoRa.available()&& i <= LenghtInput) {
       input[i] = LoRa.read(); // запис пакету до буфера
       i++;
    }
}

void JsonPacket(void) {
JSONVar JsObjectmix = JSON.parse(input);  //зміна формату буфера на строку
jsonString = JSON.stringify(JsObjectmix);

  Serial.println(input);
  JSONdemix();

 JsObject = JSON.parse(jsonString3);

// перевірка цілісності даних
if (JSON.typeof(JsObjectmix) == "undefined") {
    Serial.println("Parsing input failed!");
    return;
  }
  Serial.print("JsObject decode: ");
  Serial.println(JsObject);
 dataload=1;
     }
```

```
void JSONdemix() {

 String sedit3 = jsonString;
 char key3[50];
 String keyf3    = "0123456789KLMNOPQRSTUVWXYZ"; // ключ розшифровки
 String keyprot3 = "8372140965KLMNOPQRSTUVWXYZ";  // маска ключа

for (int i = 0, n = jsonString.length(); i < n; i++)
   {
     key3[i] = jsonString[i];
     //Serial.println(jsonString[i]);

       for (int j = 0; j < 26; j++)
       {
          if (key3[i] == keyprot3[j]) //порівняння відповідності
          {
             sedit3 [i] = keyf3[j];
          }
       }
   }
 jsonString3 = sedit3;
 //Serial.println(jsonString3);
}
```

Даний код виконує перевірку на наявність вхідного повідомлення, після чого зчитує його, звертається до функції розшифрування JSONdemix(), зберігає ключ в пам’ять та перемикає пристрій в режим читання карток.

Наступна функція використовується для визначення змінної яку вводить користувач до серійного порту.

```
// ########################################
//serial_input()
void serial_input(void) {
   char temp = Serial.read();  // зчитуваннння змінної
   if(temp == 'L' || temp == '1')  // порівняння змінної
```

```
     { dataload=0;  }  // задання значення змінної для преходу між режимами
    else if(temp == 'R' || temp == '2')
     {dataload=1;}   // задання значення змінної для преходу між режимами
    else if(temp == '-' || temp == 'z')
     {last_card_show();}  // показаит останню зчитану карту
    else if(temp == 'O' || temp == 'k')
     {key_chek();} // перехід до функції перевірки ключа
  }
// ##########################################


void key_chek(void){

 Serial.print("\n Card UID:");
if(BlockSize==0){
  Serial.print(" *No Data*");
  }

for (byte j = 0; j < BlockSize; j++) {
 truekey = 0;
 Serial.print("\n RC522 =  ");
 JSONVar JsObject1 = dataBlock[j];// 0;
 Serial.print(dataBlock[j]);
 Serial.print(" LoRa =  ");
 Serial.print(JsObject[j]);

if(JsObject1 == JsObject[j]){truekey = 1;} // посимвольне порівняння ключів
  else{return;}  // завершення функції у раз неспівпадіння
 Serial.print(" TK =  ");
 Serial.print(truekey);
    }
    open_Door();    //подача сигналу на реле замка
  }

// ################################################
void open_Door(void) {
  digitalWrite(Dor, HIGH);
   for (int Dtime = 0; Dtime < 15; Dtime++){
```

```
  delay(1000);
  }
  digitalWrite(Dor, LOW);
}
// ##################################################
```

Функція key_chek використовується для порівняння ключа з пам’яті з ключем зчитаним за допомогою модуля RC-522.

Функція open_Door() використовується для відкриття реле на певний час.

```
// ##################################################
void MFRC522doing(void) {

  MFRC522::MIFARE_Key key;
  for (byte j = 0; j < 6; j++) key.keyByte[j] = 0xFF;

  MFRC522::StatusCode status;
 if ( mfrc522.PICC_IsNewCardPresent()) {
  if (  mfrc522.PICC_ReadCardSerial()) {
    card_readv2();
     }}
}

void card_readv2(void) {
    Serial.print("\n\n**Card Detected:**");
    Serial.print("\nCard UID:");
    BlockSize = 0;
    dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
    Serial.print("\nPICC type: ");
MFRC522::PICC_Type piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);
    Serial.print(mfrc522.PICC_GetTypeName(piccType));
    Serial.print("\n**End Reading**\n");
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
  key_chek();
}
```

Функція MFRC522doing() використовується для визначення наявності ключа біля зчитувача та запускає зчитування ключа card_readv2(). Код виводить значення зчитаного ключа та зберігає його до буферу, який використовується як журнал ключів.

```
void dump_byte_array(byte *buffer, byte bufferSize) {
   BlockSize = bufferSize;
   for (byte j = 0; j < bufferSize; j++) {
      dataBlock[j] = buffer[j];
      Serial.print(buffer[j] < 0x10 ? " 0" : " ");
// запис у форматі шіснадцятизначного числа
      Serial.print(buffer[j], HEX);
   }
}
```

Функція dump_byte_array() використана для запису ключа у шіснадцятковому форматі та затису його у буфер dataBlock[] посимвольно.

Функція last_card_show() викликається за допогою серійного порту та повертає до нього останню карту, яку зчитав ID зчитувач, в незалежності від того, чи є даний ключ в списку дозволених ключів.

```
// ##############################################
void last_card_show(void) {
Serial.print("\n Card UID:");
if(BlockSize==0){
  Serial.print(" *No Data*");  //вивід повідомлення якщо дані відсутні
  }
for (byte j = 0; j < BlockSize; j++) {
      Serial.print(dataBlock[j] < 0x10 ? " 0" : " ");
      Serial.print(dataBlock[j], HEX);
   }
}
```

3.3 Перевірна працездатності

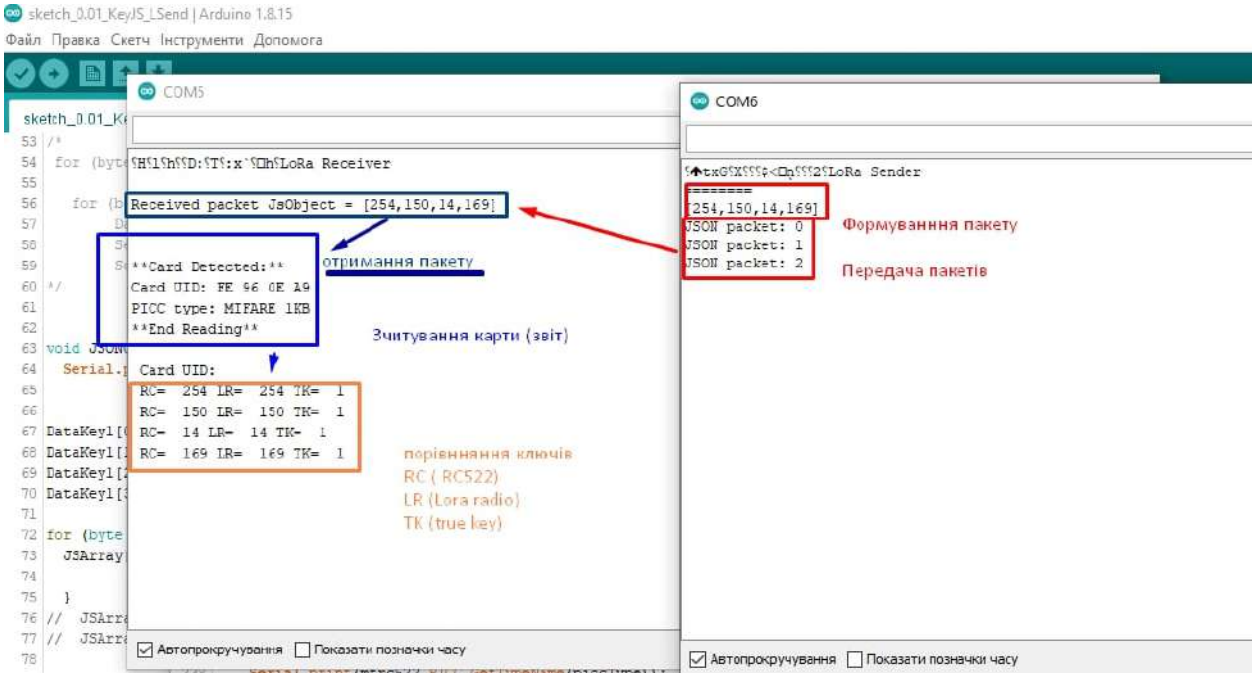

Рис. 22 результат виконання

При включенні головного пристрою, він формує пакет (рис.22 COM 6), та починає цикл передачі даних. Виконавчий пристрій (рис.22 COM 5) після включення одразу переходить в режим отримання пакетів. Отримує 1 пакет, робить його обробку та зберігає в свою базу, після чого переходить в режим читання карток. Інформація про зчитану карту виводиться після її зчитування, та порівнюється з наявним ключем. Якщо всі значення співпали — виконується фукнція відкривання дверей.

ВИСНОВОК

В ході виконання дипломного проекту було зроблено наступні висновки:

1. Проведено аналіз актуальності тематики та проведено наліз існуючих рішень на ринку, для підсумування їхніх основних характеристик для їх переносу на платформу з використанням мікроконтролера.

2. Розглянуто формати передачі даних та обрано найкращий спосіб предачі для використання в системі декількох пристроїв системи.

3. Сформовано функціонал системи контролю доступу з використанням компонентів сумісних з мікроконтролерами.

4. Обрано програмне забезпечення, з використанням якого було спроектовано систему безключового доступу.

5. Здійснено тестування прототипу системи та перевірка її функціоналу.

6. Отримано систему просту систему контролю доступу, яка готова до впровадження.

ПЕРЕЛІК ПОСИЛАНЬ

1. Physical Security Access Control Systems [Електронний ресурс]. — Режим доступу: https://www.greetly.com/blog/physical-security-access-control-systems
2. Access control [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Access_control#Electronic_access_control
3. Система безключового доступу в машину [Електронний ресурс]. — Режим доступу: https://avtotachki.com/uk/sistema-besklyuchevogo-dostupa-avto/
4. Імобілайзер [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Імобілайзер
5. Система обліку IN01 [Електронний ресурс]. — Режим доступу: https://idterminal.com/ua/p2956176-sistema-ucheta-rabochego.html
6. Dahua DHI-ASC1202B-S [Електронний ресурс]. — Режим доступу: https://bezpeka.com.ua/ua/shop/product/kontroller-dostupa-dahua-dhi-asc1202b-s/
7. Partizan PAC-12.NET [Електронний ресурс]. — Режим доступу: https://ohrana.ua/dostup/setevoy-kontroller-skud-pac-12net.html
8. U-Prox IC A [Електронний ресурс]. — Режим доступу: https://ohrana.ua/skud/schitivateli-klyuchey/setevoy-kontroller-dostupa-u-prox-ic-a.html
9. ZKTeco C3-400 [Електронний ресурс]. — Режим доступу: https://ohrana.ua/dostup/setevoy-kontroller-zkteco-c3-400-dlya-4-dverey.html
10. Hikvision DS-K2802 [Електронний ресурс]. — Режим доступу: https://ohrana.ua/skud/setevoy-kontroller-dostupa-hikvision-ds-k2802.html

11. Dahua DH-ASR1100A [Електронний ресурс]. — Режим доступу: https://bezpeka.com.ua/shop/product/rfid-schityvatel-dahua-dh-asr1100a/
12. Sparta SR10D [Електронний ресурс]. — Режим доступу: https://rozetka.com.ua/ua/133050229/p133050229/
13. ATIS PR-01 MF [Електронний ресурс]. — Режим доступу: https://rozetka.com.ua/ua/339199345/p339199345/
14. EM-Marine U-Prox mini [Електронний ресурс]. — Режим доступу: https://ohrana.ua/schitivatel-u-prox-mini-485.html
15. Raspberry Pi Pico [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod4358-raspberry-pi-pico
16. Arduino UNO R3 (CH340) [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod2610-arduino-uno-r3-ch340
17. Плата NodeMCU [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/NodeMCU
18. Datasheet NodeMCU-V3 [Електронний ресурс]. — Режим доступу: https://www.theengineeringprojects.com/Downloads/Datasheet/NodeMCU-V3.pdf
19. Плата розробника FONA [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod1682-plata-razrabotchika-fona-na-32u4-ot-adafruit
20. Передавання даних [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Передавання_даних
21. Цифровий сигнал [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Цифровий_сигнал
22. Бездротові технології [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Бездротові_технології
23. Радіо LoRa [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/LoRa
24. Zigbee [Електронний ресурс]. — Режим доступу: https://ru.wikipedia.org/wiki/Zigbee

25. Радіо LoRa на чіпі SX1278 [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod3299-modem-lora-na-chipe-sx1278
26. Datasheet LoRa SX1278 [Електронний ресурс]. — Режим доступу: https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf
27. Сканер штрих/QR-кодів [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod3012-skaner-shtrih-kodov-qr-kodov-1d2d-ot-waveshare
28. Сканер відбитків пальців [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod2622-modyl-yomkostnogo-skanera-otpechatkov-palcev
29. RC522 [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod649-rfid-modyl-rc522-s-kartochkoi-dostypa-dlya-arduino
30. RFID [Електронний ресурс]. — Режим доступу: https://ru.wikipedia.org/wiki/RFI
31. Зарядний пристрій [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod2050-zaryadnoe-ystroistvo-5v-1000ma-2xusb-type-a
32. microUSB-USB [Електронний ресурс]. — Режим доступу: https://arduino.ua/prod809-usb-2-0-kabel-microusb-usb-1-8m-atcom-c-ferritom
33. Arduino IDE [Електронний ресурс]. — Режим доступу: https://www.arduino.cc/en/software
34. C [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/C_(мова_програмування)
35. Serial Peripheral Interface [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Serial_Peripheral_Interface
36. ISM [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/ISM

37. JSON [Електронний ресурс]. — Режим доступу:
https://uk.wikipedia.org/wiki/JSON