Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет радіоелектроніки та телекомунікацій
(повне найменування інституту, факультету)

Кафедра інформаційних технологій електронних засобів
(повне найменування кафедри)

Пояснювальна записка

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

на тему Модуль контролю доступу до ліфтів

Виконав: студент(ка) 4 курсу, групи РТ-618сп

Спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
(код і найменування спеціальності)

Освітня програма (спеціалізація)
«Автоматизація, мехатроніка та робототехніка»

Каулько І. І.
(прізвище та ініціали)

Керівник Шевченко О. С.
(прізвище та ініціали)

Рецензент Мороз Г. В.
(прізвище та ініціали)

2021

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет Інститут інформатики та радіоелектроніки, Факультет радіоелектроніки та телекомунікацій
Кафедра Інформаційних технологій електронних засобів
Ступінь вищої освіти бакалавр
Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»
(код і найменування)
Освітня програма (спеціалізація) Автоматизація, мехатроніка та робототехніка
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ
Завідувач кафедри ПІБ Огренич Є.В., к.т.н.
«28» 05 2021 року

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

Каулько Інна Ігорівна
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Модуль контролю доступу до ліфтів
керівник проєкту (роботи) Шевченко Олексій Станіславович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «26» квітня 2021 року №159
2. Строк подання студентом проєкту (роботи) 7 червня 2021 року
3. Вихідні дані до проєкту (роботи) структури модулів вибору поверхів у ліфтовому обладнанні багатоповерхових житлових будинків України
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд області розробки, постановка задач кваліфікаційної роботи, розробка структури модулю, вибір елементів, розробка конструкції модулю, розробка програмного забезпечення
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) 25 рисунків, 14 таблиць, 18 слайдів

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1 - 3	Шевченко О.С., ст. викладач		
Нормоконтроль	Поспеєва І.Є., ст. викладач		

7. Дата видачі завдання «26» квітня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Огляд систем автоматизації доступу в ЖКГ	26.04.21	
2	Постановка технічного завдання	28.04.21	
3	Розробка схеми та конструкції модулю	29.04.21	
4	Розробка бази даних та протоколу обміну	03.05.21	
5	Написання тексту програмного забезпечення	29.05.21	
6	Оформлення ПЗ та захист дипломного проекту	01.06.21	

Студент(ка) _____ Каулько І.І.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи) _____ Шевченко О. С.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

ПЗ: 64 с., 14 табл., 25 рис., 9 джерел, 18 слайдів.

Об'єкт розробки: системи контролю та обліку в сфері ЖКГ.

Предмет розробки: електронний модуль з програмним забезпеченням для контролю доступу до ліфтів багатоповерхових житлових будинків.

Мета роботи: вибір компонентів, розробка структури та конструкції модуля контролю доступу до ліфтів з програмним забезпеченням для віддаленої диспетчирізації.

Для здійснення поставленої мети необхідно було вирішити наступні задачі:

- ознайомитись з особливостями організації систем контролю доступу в ЖКГ;

- провести розробку структури модульної системи контролю доступу в ліфти багатоповерхових житлових будинків;

- розробити конструкцію модулю контролю доступу в ліфти;

- розробити базу даних для зберігання даних, щодо права доступу в ліфти;

- розробити протокол обміну модуля доступу в ліфти з іншими елементами системи доступу;

- розробити програму керуючого мікроконтролеру модуля доступу в ліфти.

МОДУЛЬ, ПЛАТА, КОНСТРУКЦІЯ, ПРОГРАМА, СЕРВЕР, БАЗА ДАНИХ, ЛІФТ, ОСББ, ПРОГРАМУВАННЯ, СИСТЕМА КОНТРОЛЮ ДОСТУП

ЗМІСТ

Вступ

Системи та модулі контролю доступу останнім часом набувають все більшої популярності завдяки необхідності обмеження проникнення на території та обмеження користування окремими об’єктами особами, що не виконують фінансових розрахунків.

Під системою контролю доступу можна розуміти все, що обмежує потрапляння чужих на територію. У самому звичному вигляді це охоронці (але це людський фактор який ще потребує великих поточних витрат). Більш сучасний варіант передбачає автоматизацію. Контроль проходу людей - це турнікети і двері і хвіртки з електромеханічними і електромагнітними замками. Ці системи актуальні не лише для виробничих та офісних будівель, а й для житлових багатоквартирних будинків.

Часто виникає питання обмеження доступу до ліфтів у багатоквартирних житлових будинках пов’язане з обмеженням доступу стороннім особам та у якості стимулювання сплати за обслуговання будинку. Саме сплата за обслуговання будинку для ОСББ та керуючих компаній є дуже болючим питанням. Оскільки якщо мешканець не платить за світло, воду або газ – йому відключають постачання цього ресурсу, а у випадку зі сплатою за обслуговання будинку все набагато складніше. Голова ОСББ не може припинити обслуговувати окрему квартиру адже послуги надаються для всіх жильців дому (ремонт мереж, прибирання, вивіз сміття і т.і.), а отже припинивши надавати одну з цих послуг керуюча компанія або керівництво ОСББ порушить право тих хто сплачує. Адресна можливість користування ліфтом лише для тих, хто сплачує за обслуговання будинку частково може вирішити проблему неплатежів (особливо стосовно боржників з високих поверхів).

Для чого потрібен контроль доступу для ліфтів:

- обмеження числа осіб, які можуть користуватися ліфтом. Таке рішення стає ефективним заходом проти хронічних неплатників за послуги ліфтових

операторів і ЖКГ. Одночасно перекривається доступ до ліфту громадянам, які не проживають у даному будинку;

- обмеження доступу в технічні приміщення, призначені для обслуговування ліфтового обладнання.

- розвантаження ліфтових систем в години пік;

- зниження актів вандалізму і кримінальних злочинів, скоєних в ліфтових кабінах

Існує можливість збору інформації про користувачів з відображенням і веденням історії на диспетчерському пульті управління персональних даних особи, яка в даний момент викликає ліфт або знаходиться в кабіні. Обмеження маршрутів користувачів конкретними поверхами - ще одна поширена опція, яка користується особливою популярністю в комерційних і споруджуваних будинках.

В кваліфікаційній роботі пропонується розробка модулю контролю доступів для ліфтів, що дозволить віддалено адмініструвати доступ жильців багатоквартирних домів, що надасть можливість знизити заборгованість за обслуговання будинків та зменшити амортизацію ліфтового обладнання.

Об'єкт розробки: системи контролю та обліку в сфері ЖКГ.

Предмет розробки: електронний модуль з програмним забезпеченням для контролю доступу до ліфтів багатоповерхових житлових будинків.

Мета роботи: вибір компонентів, розробка структури та конструкції модуля контролю доступу до ліфтів з програмним забезпеченням для віддаленої диспетчирізації.

1 Огляд області розробки та поставка задач

1.1 Системи контролю доступу в ЖКГ

Система контролю доступу до ліфта насправді є технологією управління ліфтом. Системний пристрій, схожий на систему контролю доступу встановлену на оригінальному ліфті. Тільки після того, як карта буде проведена, ліфт запуститься, і власник буде відправлений на поверх призначення. Системи контролю доступу ліфта використовують безконтактні картки в якості облікових даних для входу і виходу з будівлі.

У міру прогресу і розвитку суспільства спосіб життя людей зазнав глибокі зміни. Виникнення міських висотних будівель зробило життя людей більш просторою. Інтелектуальна система контролю доступу до ліфтів - це високотехнологічний продукт, що відповідає вимогам нашої епохи. Використання системи контролю доступу до ліфтів значно поліпшило інтелектуальне будівництво та управління будівлями, і в той же час максимально знизило занепокоєння власників. Його легко об'єднати з іншими інтелектуальними системами в більш потужну і всеосяжну систему, яка підходить для розширеного управління різними інтегрованими методами.

Ліфтове устаткування є зоною підвищеної уваги, в якій постійно виникають різні ризики, які несуть загрозу для здоров'я і життя людини. Одночасно ліфтові кабіни найчастіше стають місцем для проявів актів вандалізму і дій кримінального характеру. А проникнення сторонніх осіб в технічні периметри загрожує найнесподіванішими і неприємними наслідками.

Саме тому в сучасних ліфтових господарствах часто використовуються системи контролю та управління доступом (СКУД) система, що дозволяють розділити повноваження і зони відповідальності.

На сьогоднішній день, ймовірно, немає багатоквартирних домогосподарств, які не зіткнулися з проблемою неплатежів за внесками на утримання будинку та прибудинкової території. Проблема зростаючих боргів

не тільки перешкоджає нормальному (нормативному) обслуговуванню будинку, а й унеможливлює накопичення коштів на ремонт і модернізацію будинку і його інженерних систем. Фактично "ремонтними" грошима, а по суті за рахунок коштів сумлінних мешканців, закриваються створювані боржниками "дірки" в загальнобудинкових бюджеті і цей процес циклічний як замкнуте коло.

На жаль, в більшості випадків заклики адміністрацій будинків до совісті боржників, оголошення списків і вивішування дощок ганьби не мають дієвих результатів, а судові тяжби обтяжливі більш для управителя, ніж для несумлінного мешканця. Неплатник для себе давно усвідомив, що в умовах перманентної нестабільності економіки і знецінення національної валюти, його борги будуть з часом також знецінені.

Отже для стимулювання боржників гасити оплату за обслуговання будинку актуально впровадити систему доступу до ліфтів, що дозволить обмежити проїзд жильців, що мають заборгованість. При цьому права жильця не порушуються – він може вільно пересуватися будинком але доведеться ходити пішки, що особливо актуально для жителів верхніх поверхів.

Призначення систем контролю доступу для ліфтів:

- обмеження числа осіб, які можуть користуватися ліфтом. Таке рішення стає ефективним заходом проти хронічних неплатників за послуги ліфтових операторів і ЖКГ. Одночасно перекривається доступ до ліфту громадянам, які не проживають у даному будинку;

- обмеження доступу в технічні приміщення, призначені для обслуговування ліфтового обладнання.

- розвантаження ліфтових систем в години пік;

- зниження актів вандалізму і кримінальних злочинів, скоєних в ліфтових кабінах

1.2 Огляд аналогів

Перш ніж перейти до власної розробки необхідно провести аналіз аналогів, що існують на ринку.

Ліфтова система доступу ZKTeco.

Ліфтова система доступу ZKTeco забезпечує можливість доступу на контрольовані поверхи тільки авторизованим персонам. Для можливості виклику потрібного поверху користувач повинен спершу пройти перевірку за безконтактною карті, відбитком пальця або кодом доступу. Таким чином здійснюється другий рубіж системи безпеки всієї будівлі, відвідувачі, які мають право доступу в будівлю можуть відвідати тільки цільові поверхи.

ZKTeco був розроблений спеціалізований контролер EC10 для обмеження доступу на поверхи через ліфт. Принцип роботи системи доступу на поверх полягає в тому, що користувач, зайшовши в кабіну ліфта, прикладає палець або пред'являє зчитувача ідентифікатор. Якщо користувач успішно розпізнано і має необхідні права, після натискання на кнопку потрібного поверху - ліфт зупиниться на необхідному рівні.

Залежно від обраної конфігурації зчитувача, у адміністратора системи є можливість обмежити доступ за відбитком пальця, або безконтактної картки, або призначити кожному користувачеві індивідуальний ПІН-код. Доступні різні сценарії контролю доступу: доступ за розкладом, режим вільного доступу

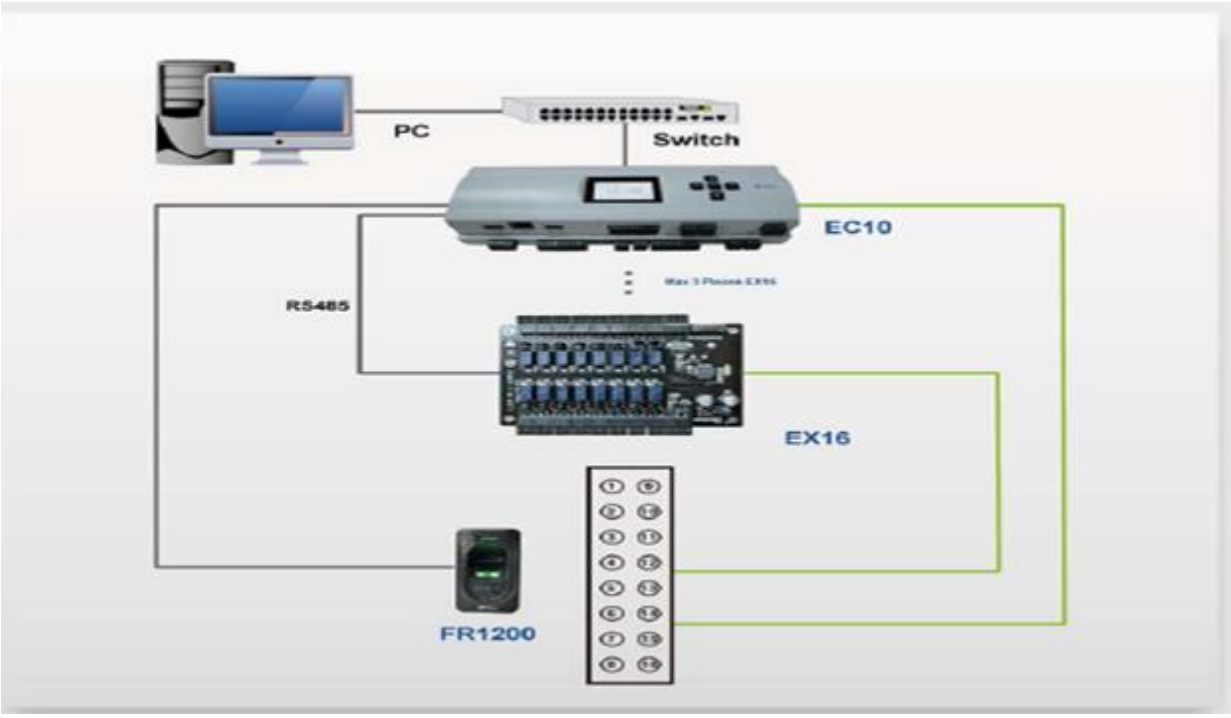


Рисунок 1.1 –Состав лифтовой системы ZKTeco

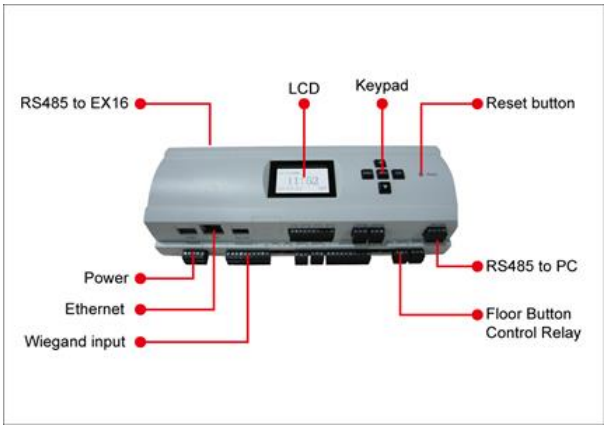


Рисунок 1.2 – EC10- контроллер керування ліфтами (керування до 10 поверхів).

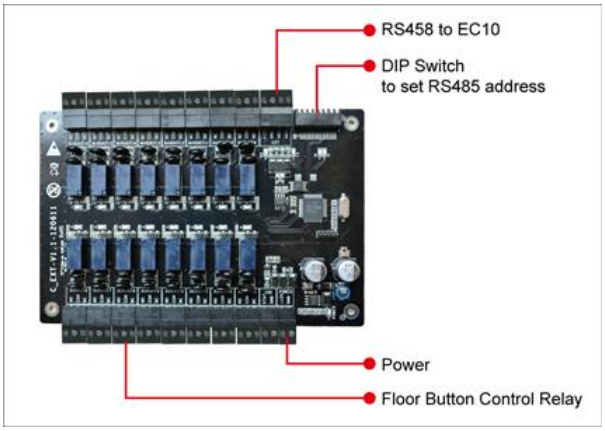


Рисунок 1.3 – EX16- плата розширення (керування до 16 поверхів).

Особливості системи контролю доступу управління ліфтами:

Підтримка безконтактних і біометричних зчитувачів. Використання rfid зчитувачів серії KR або біометричного сканера FR1200.

Режим безвідмовного доступу. При виході з ладу системи, всі користувачі матимуть доступ на всі поверхи.

Завантаження прав доступу в реальному часі. Після введення прав доступу до програми, інформація в контролер буде передана негайно.

Моніторинг подій в реальному часі. Події доступу відображаються в програмі в реальному часі.

Гнучка настройка прав доступу. Користувач може мати доступ на один або кілька поверхів.

Система контролю доступу "ПОРТАЛ"

Система контролю доступу "Портал" призначена для організації розмежування доступу в ліфтові кабіни, об'єкти, що охороняються, приміщення. Система відрізняється легкістю контролю і управління, а також можливістю нарощування модулів.

СКД «Портал» функціонує як розподілена система контролю і управління. Запропонована система дозволяє домогтися максимальної гнучкості системи, простоти управління і обслуговування, а також можливості легкого розширення.

СКД «Портал» має функцію захисту від копіювання ключів. Для реалізації даної функції необхідно застосування ключів типу DS1971.

Система працює з різними типами ключів:

- контактні ключі: DS1990, DS1971;

- безконтактні ключі (карти) стандарту EM-Marin;

- безконтактні ключі (карти) стандарту Mifare;

Система контролю доступу "Портал" виконує наступні функції:

- розмежування доступу на об'єкт (кабіна ліфта, об'єкти, що охороняються, приміщення);

- дистанційний контроль стану об'єкта;

- дистанційне керування;

- автоматичне керування;

- ручне керування.

Бувають автономні системи управління з карток для ліфта. Вони вбудовані в кнопки керування ліфта і програмуються з використанням майстер карти.

Рисунок 1.4 – Модуль SCHAEFER ES42

Як видно на фотографії вони виконані у вигляді кнопки від ліфта і тому відрізнити від кнопки, якщо контролер вже встановлений, то практично неможливо. Контролери надійні і довговічні не вимагають додаткового обслуговування.

Але мають ряд недоліків: 1) Програмування ключів в контролер робиться за допомогою майстер карти. 2) кількість ключів в пам'яті не дуже велике. 3) Видалення ключів теж не просте заняття.

Недоліками розглянутих аналогів є дротова побудова системи, а також необхідність постійного перепрограмування ключів доступу та ліфтового обладнання, а отже потребує безпосереднього виїзду або спеціаліста обслуговуючої (керуючої) компанії в будинок або виїзд (прихід) користувачів до офісу керуючої компанії для поповнення кількості поїздок або розблокування ключів доступу.

Недоліки аналогів:

- для внесення нових ключів та видалення втрачених необхідно перепрограмування модулю спеціалістом компанії, що потребує виїзду спеціалісту к ліфту;

- внесення обмеженої кількості поїздок – оплата за кожну поїздку – жильцю поїздки треба поповнювати відвідавши представника керуючої компанії;

- дротове з’єднання модулю з локальною мережею або з інтернет у разі мережевої структури;

- відсутність віддаленої диспетчирізації та можливості внесення змін у перелік ключів та прав на проїзд.

1.3 Постановка задач

На базі аналізу існуючих систем керування доступу до ліфтів та загальної проблематики було запропоновано вирішити наступні задачі проектування в кваліфікаційній роботі бакалавра.

Пропонована система дозволятиме проводити контроль користування ліфтом в під'їзді багатоповерхового будинку.

Система дозволятиме:

- проводити контроль часу і кількості користування ліфтом мешканцями;

- віддалено обмежувати доступ до користування неплатниками.

Переваги в порівнянні з аналогами:

- немає необхідності відвідування керуючої компанії для активації нової карти або прописування кількості поїздок;

- віддалений облік і контроль з будь-якої точки планети;

- для внесення змін в список санкціонованих користувачів досить внести зміни у запис у програмному забезпеченні;

- резервування даних на віддаленому сервері без можливості їх втрати;

- вбудована пам'ять на випадок збоїв з мережею інтернет.

Принцип роботи системи.

Мешканці під'їзду отримують (базовий пакет з 36 карт або брелоків по вибору) та додатково купують у разі коли на квартиру необхідно більш ніж

один ключ (вартість одного брелока або однієї карти на вибір 20 грн) RFID брелок і / або RFID карту.

Унікальний ідентифікаційний номер карти вводиться в програму із зазначенням даних користувача (ПІБ мешканця, номера квартири в якій він проживає). Керуюча компанія, ОСББ або будь-яке інше відповідальна особа, яка розподіляє доступ до ліфта вказує список користувачів, яким дозволений доступ. Ці дані відправляються на віддалений сервер для зберігання. Завжди можна переглянути журнал змін, що проводяться і списки санкціонованих користувачів.

Мешканець при вході в ліфт прикладає карту або брелок до RFID зчитувача і якщо доступ даному мешканцю дозволений мікропроцесорний модуль управління розблокує клавіатуру ліфта і мешканець може скористатися ліфтом.

При прикладанні карти / брелока модуль управління по бездротовій мережі зв'язується з віддаленим сервером, на якому зберігатися списки санкціонованих карт і отримує дозвіл або заборону на доступ до ліфта. У разі якщо в даний момент по якійсь причині не можуть зв'язатися з сервером (відсутнє інтернет з'єднання) список отримується з вбудованої пам'яті. Вміст пам'яті оновлюється кожного разу при підключенні до сервера через вузол збору інформації по території.

Принцип дії безконтактної RFID картки (брелока) заснований на радіосигналі. Це означає, що для коректної роботи такої картки (брелока) механічний контакт зі зчитувачем картоприймача не потрібен. Пристрій спрацьовує при піднесенні безконтактної картки (брелока) до передньої панелі картоприймача. У багатьох випадках для спрацьовування картоприймача досить піднести картку або гаманець, де зберігається картка, не виймаючи саму картку.

Безконтактна картка не боїться подряпин і інших не руйнуючих фізичних впливів.

Якщо користувач бажає носити свій електронний ключ на одній зв'язці

зі звичайними ключами, можна використовувати безконтактний брелок.

Також як додаткову можливість пропонується використовувати вмонтований в більшість сучасних смартфонів NFC модуль, що співпадає по частоті роботи з RFID 13,56МГц та має повністю ідентичний принцип авторизації. Це дозволить зовсім відмовитися від використання додаткових ключів доступу якщо є смартфон з відповідною функцією.

2 Розробка схеми та конструкції модулю

2.1 Розробка структурної схеми модулю

На першому етапі розробки модулю була розроблена загальна схема системи контролю доступу наведена на рис.2.1.

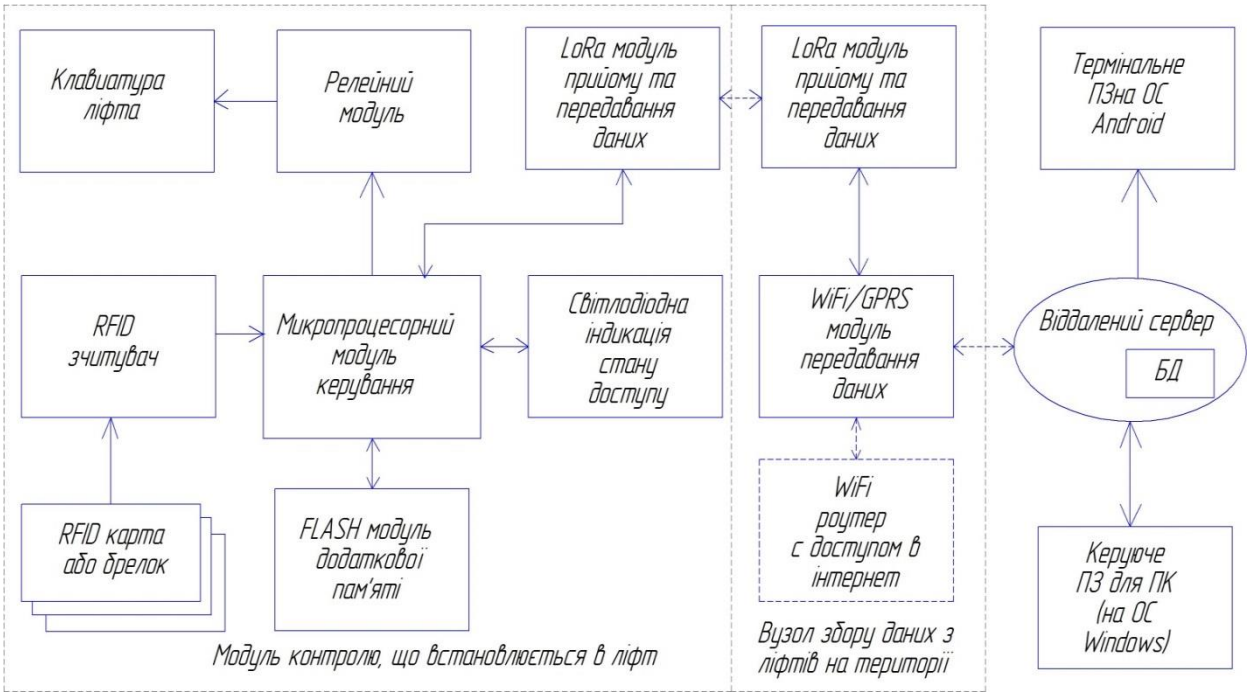


Рисунок 2.1 – Структурна схема системи доступу в ліфти

Система складається з п’яти вузлів:

- модуль контролю, що встановлюється безпосередньо у ліфт;

- вузол збору даних з ліфтів на певній території;

- віддалений сервер з базою даних для зберігання інформації о користувачах, картах доступу і т.і.;

- термінальне ПЗ на ОС Android для перегляду даних щодо стану доступу та переліку ключів;

- керуюче ПЗ для ПК у вигляді додатку, що встановлюється на ОС Windows або у вигляді WEB-додатку.

Модуль контролю, що встановлюється у ліфт керує дозволом на проїзд пасажирів. Пасажир входячи в кабіну ліфта прикладує до RFID зчитувача

власну картку або брелок, який йому попередньо видано керуючою компанією або головою ОСББ. Унікальний ідентифікаційний модуль зчитується та аналізується мікропроцесорним модулем. Якщо доступ для цього ідентифікаційного модуля (картки або брелока) дозволений – замикається реле (нормально розімкнене) яке подає живлення на клавіатуру ліфта та пасажир має змогу натиснути кнопку поверху на який збирається їхати.

Перевагою системи, що розробляється є можливість віддаленого внесення, редагування та видалення ключів доступу, надання дозволу або вмикання заборони на проїзд окремим жителям будинку в залежності від сплати за комунальні послуги та поводження у ліфті (забороняти проїзд тим хто замічений у вандалізмі щодо ліфтового обладнання).

Для виявлення права проїзду за карткою модуль керування доступом при читанні ключа доступу перевіряє його наявність у внутрішній енергонезалежній пам’яті, а також помітку про дозвіл на проїзд. Якщо ключ внесений у пам'ять та дозвіл проїзду за цим ключем дозволений модуль через реле вмикає живлення клавіатури вибору поверху та пасажир може скористатися роботою ліфта.

Для внесення змін до переліку карток та дозволів на проїзд модуль зв’язується з вузлом збору даних з ліфтів по території з допомогою бездротової мережі та якщо вузол повідомить модуль про наявність змін у переліку або правах доступу модуль отримає оновлений перелік з дозволами та запише у FLASH пам'ять.

У свою чергу вузол збору даних отримує інформацію щодо переліку ключів та дозволів від віддаленого серверу з базою даних за допомогою інернет з’єднання (бездротово через WiFi, GPRS або через кабель по Ethernet).

Адміністратори системи та уповноважені особи керуючих компаній або ОСББ за допомогою термінального ПЗ на ПК та/або WEB інтерфейсу можуть вносити зміни до переліку ключів, що мають доступ. Ці зміни будуть збережені на сервері у базі даних та будуть передаватися на територіальні вузли, а далі на модулі контролю доступу до кожного ліфта.

Окремі користувачі за допомогою мобільного додатку та WEB інтерфейсу зможуть передивитися стан доступу по своїй адресі проживання.

У рамках нашої кваліфікаційної роботи ми зупинилися на розробці модулю контролю, що встановлюється безпосередньо у ліфт оскільки в нас обмежений час виконання.

Для передавання даних від ліфтів до вузлу збору даних та у зворотньому напрямку було обрано технологію LoRa а для безпосередньої передачі протокол LoRaWAN.

В основі технології LoRa лежить однойменний метод модуляції, який був запатентований компанією Semtech. Цей метод ґрунтується на принципі розширення спектра і лінійної частотної модуляції. В процесі передачі дані кодуються широкосмуговими імпульсами з зменшується або збільшується, в певному часовому діапазоні, частотою. Дане рішення дозволяє зробити приймач стійким до відхилень частоти від номінального значення, що знижує вимоги до якості генератора і дозволяє використовувати прості кварцові резонатори. За рахунок використання технології розширення спектру, приймач LoRa може демодулировать сигнал, який має рівень на 20 дБ рівня шумів. Висока чутливість приймачів (-148 дБм) дозволяє використовувати застосовувати дану технологію на великих відстанях, забезпечуючи мале енергоспоживання і високу стійкість зв'язку.

LoRaWAN - відкритий протокол зв'язку, який визначає архітектуру системи. Цей протокол передбачає топологію типу «зірка». LoRaWAN розроблявся з метою організації зв'язку між недорогими пристроями, які можуть працювати від батарей (акумуляторів). Для забезпечення прийнятного відносини швидкості передачі до енергоспоживання, протокол передбачає різні класи вузлів. Протокол LoRaWAN визначає конкретний набір швидкостей передачі даних, але реалізація фізичного рівня моделі OSI буде залежати від обраної мікросхеми.

На відміну від великого числа існуючих мереж, що використовують mesh-архітектуру, де вузли мережі, для розширення покриття передають

інформацію від одного до іншого, LoRa - мережа використовує топологію «зірка». Це дозволяє зменшити енергетичні споживання пристроїв (за рахунок відсутності необхідності пересилання пакетів від інших пристроїв), і спростити архітектуру мережі.

У мережі LoRaWAN вузол пов'язується не з конкретним шлюзом, а передає дані на кілька шлюзів. Кожен шлюз пересилає отриманий пакет від кінцевого вузла через транспорт (мережа, Wi-Fi, Ethernet або інше) на хмарний сервер. Сервер управляє мережею, відкидає надлишкові пакети, виконує перевірки безпеки, планує оптимальний маршрут передачі підтверджує повідомлення та управляє швидкістю передачі даних. Використання такої архітектури дозволяє позбутися від процедури хендовера при переміщенні мобільних датчиків в межах дії мережі. Вузли в мережі працюють в асинхронному режимі і передають дані в міру накопичення або по перериванню.

Для доступу до ресурсів мережі використовується метод Aloha. Відмова від постійної синхронізації пристроїв (як в mesh- або стільникових мережах) так само дозволяє економити заряд батареї. У мережі з «зоряної» топологією складно організувати велику ємність мережі одночасно з великою площею покриття. Для реалізації такої можливості в LoRaWAN застосовують адаптивну швидкість передачі даних і використовують багатоканальні мульти модемні трансивери в шлюзах, щоб повідомлення могли передаватися одночасно по декількох каналах. Критичні фактори для пропускної здатності - кількість одночасних каналів, швидкість передачі даних (час в ефірі), довжина корисного навантаження і наскільки часто вузли ведуть передачу.

Оскільки LoRa є модуляцію на основі розширення спектра, сигнали практично ортогональні один одному, коли використовуються різні коефіцієнти розширення. При зміні коефіцієнта розширення ефективна швидкість передачі даних теж змінюється. Шлюз використовує це властивість, маючи можливість отримувати кілька різних швидкостей передачі даних на одному каналі одночасно. Якщо вузол має добре сполучення і знаходиться

близько до шлюзу - він може використовувати більш високу швидкість передачі даних, при цьому час його перебування в ефірі стає менше, що відкриває «вікно» для передачі від інших вузлів. Мережа може будуватися виходячи з необхідної ємності, наприклад можна збільшити потужність в мережі, встановивши більшу кількість шлюзів або зменшити кількість шлюзів, які будуть прослуховувати клієнти і збільшити пропускну здатність каналу в 6-8 разів.

2.2 Вибір елементів

У якості мікроконтролерного модулю було обрано модуль Arduino Nano – він має невеликий розмір та ціну при тому забезпечує необхідну кількість портів вводу-виводу та має достатню пам'ять програм, оперативну пам'ять та пам'ять даних.

Додатковою перевагою модулю є можливість його перепрограмування без застосування зовнішнього програма тора – достатньо здійснити підключення через USB шнур до роз’єму комп’ютера. Середовище розробки програмного забезпечення безкоштовне та має велику кількість безкоштовних бібліотек для різних датчиків, пристроїв виводу. Мікроконтролер на базі якого побудований модуль має відкриту архітектуру, що дозволяє створювати власні бібліотеки модулів та вносити зміни в існуючи прискорюючи та оптимізуючи при цьому кінцевий програмний код пристрою.

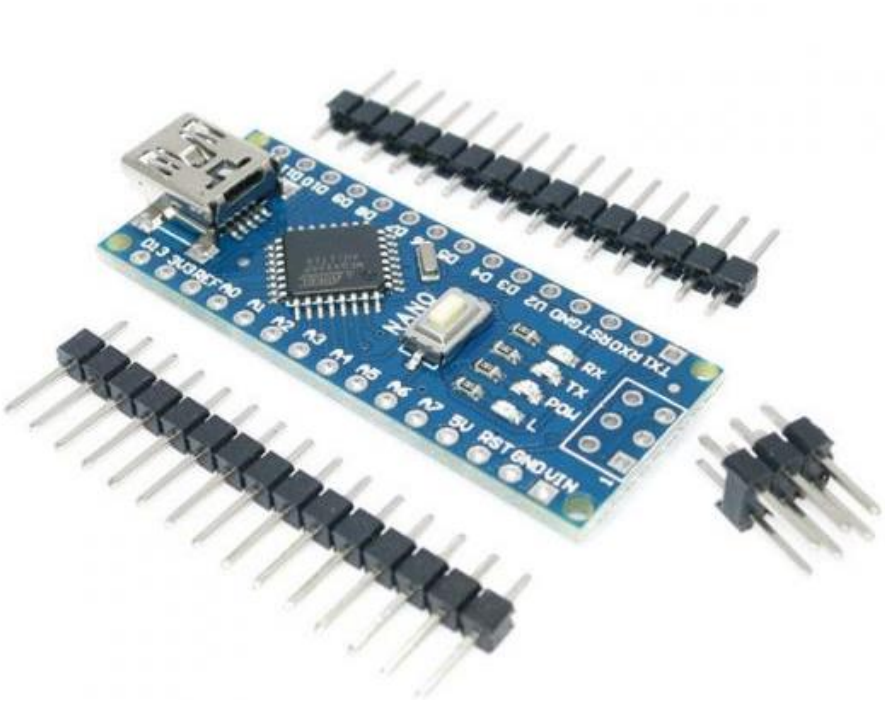


Рисунок 2.2 – Модуль Arduino Nano

У якості модуля додаткової пам’яті обрано модуль з мікросхемою флеш пам'яті W25Q64 об'ємом 64МБіт = 8 МБайт. Підключення по інтерфейсу SPI. Напруга харчування і логіки 3.3 В.

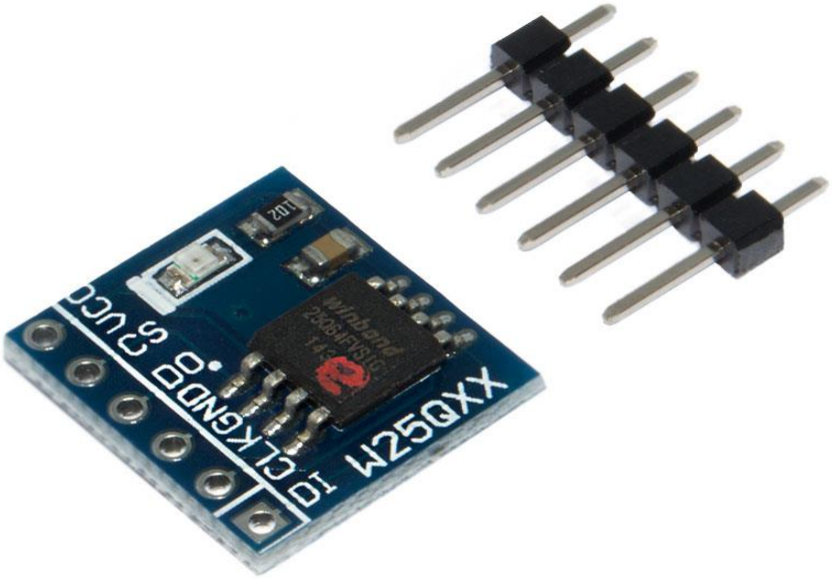


Рисунок 2.3 – Модуль W25Q64 64Mbit 8MByte FLASH DataFlash SPI

W25Q64 (64 Мбіт) - IC послідовної флеш-пам'яті, орієнтовані на застосування в складі систем з обмеженим простором, лініями вводу-виводу і малим споживанням. Серія 25Q володіє рівнем гнучкості і робочих характеристик, який перевершує IC звичайної флеш-пам'яті.

Вони ідеальні для тіньового зберігання програмного коду, який завантажується в оперативній пам’яті при подачі живлення, для вибірки коду програми (XIP) безпосередньо з флеш-пам'яті (в подвоєному режимі або в режимі з збільшенням пропускної здатності інтерфейсу SPI у чотири рази), а також для зберігання голосової, текстової та числової інформації. Інтегральні схеми працюють від однієї напруги живлення 2.7 ... 3.6 В з споживаним струмом не більше 5 мА в активному режимі і не більше 1 мкА в режимі зниження потужності. Всі інтегральні схеми представлені в мініатюрних корпусах.

У якості модуля передавання було обрано Модем LoRa на чіпі SX1278

Рисунок 2.4 – Модем LoRa 433 МГц на чіпі SX1278

Приймач SX1278 оснащений модемом дальнього радіусу дії LoRa®, який забезпечує надширокосмуговий зв'язок з розширеним спектром і високу стійкість при мінімальному споживанні струму. Використовуючи запатентовану Semtech техніку модуляції LoRa, SX1278 може досягти чутливості понад -148 дБм, використовуючи недорогі мікросхеми і супутні

комплектуючі. Висока чутливість в поєднанні з вбудованим підсилювачем потужності +20 дБм забезпечує кращий в галузі бюджет каналу зв'язку, що робить його оптимальним для будь-якого застосування, що вимагає частотного діапазону або надійності. LoRa. Модуль також забезпечує переваги як в блокуванні так і в вибірковості в порівнянні з традиційними методами модуляції, вирішуючи традиційний компромісний задум між діапазоном, завадостійкістю і споживанням енергії.

Характеристики:

- метод модуляції: FSK / GFSK, LoRa;

- тип зв'язку: напівдуплексних зв'язок;

- відхилення каналу (ADJ): 56dBm;

- чутливість приймача (RX): -139dBm;

- робочий діапазон: ISM multiband;

- інтеллектуальне скидання, монітор низької напруги, синхронний wakeup, режим низької потужності, режим сну;

- енергоспоживання в режимі прийому: 12 ~ 13 мА;

- внутрішній буфер: 256 байт FIFO TX / RX;

- виявлення сигналу каналу передачі даних: ISSI;

- режим передачі: FIFO / прямий режим (рекомендований режим пакета FIFO)

Варіанти конфігурації модуля:

- AFC;

- пробудження при наявності радіосигналу;

- знижена споживана потужність;

- виявлення несучої;

- FEC корекція помилок;

- EC шифрування.

Для замикання розмикання ланцюга живлення клавіатури вибору поверху на внутрішній панелі кабіни ліфта було обрано релейний модуль, що наведено на рис.2.5.

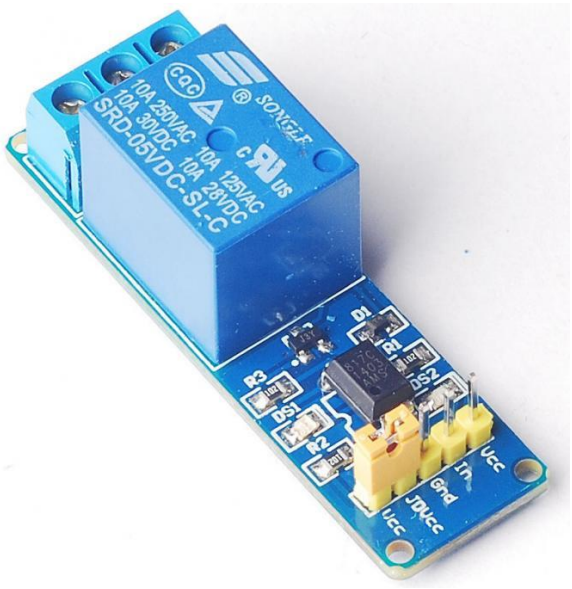


Рисунок 2.5 – Релейний модуль для замикання/розмикання живлення клавіатури вибору поверху в кабіні ліфту

У якості зчитувача RFID обрано модуль RC522, що працює на частоті 13,56МГц (рис.2.6) у першу чергу за рахунок його роботи по SPI інтерфейсу.

Модуль RC522 - RFID-модуль 13,56 МГц з SPI-інтерфейсом. Модуль RFID RC522 заснований на мікросхемі MFRC522 від NXP. Поставляється з двома мітками, RFID-карткою і брелоком з об'ємом пам'яті 1 КБ.

Модуль RFID RC522 створює електромагнітне поля з частотою 13,56 МГц, яке використовується для зв'язку з RFID мітками (стандартні мітки ISO 14443A). Для взаємодії з контролерами, модуль використовує 4-х контактний інтерфейс SPI. Так само, модуль підтримує протоколи зв'язку I2C і UART.

Рисунок 2.6 – Модуль зчитувача RC522

Після вибору елементів модулю було проведено розробку схеми електричної. Схема була розроблена в середовищі EasyEda [10] оскільки воно дозволяє проводити тестування роботи схеми з врахуванням завантаження розробленої програми керуючого мікроконтролера. Додатково це середовище безкоштовне та має онлайн редактор, що зв’язаний з аккаунтом Google, а отже проект доступний з будь якої точки планети де є інтернет.

Спочатку згідно схеми документації Arduino Nano було створено схему для чотирьох компонентів:

- мікроконтролера з обв’язкою (рис.2.7);
- перетворювача TTL-USB (рис.2.8);
- схеми живлення плати (рис.2.9);
- схеми комутації (рис.2.10).

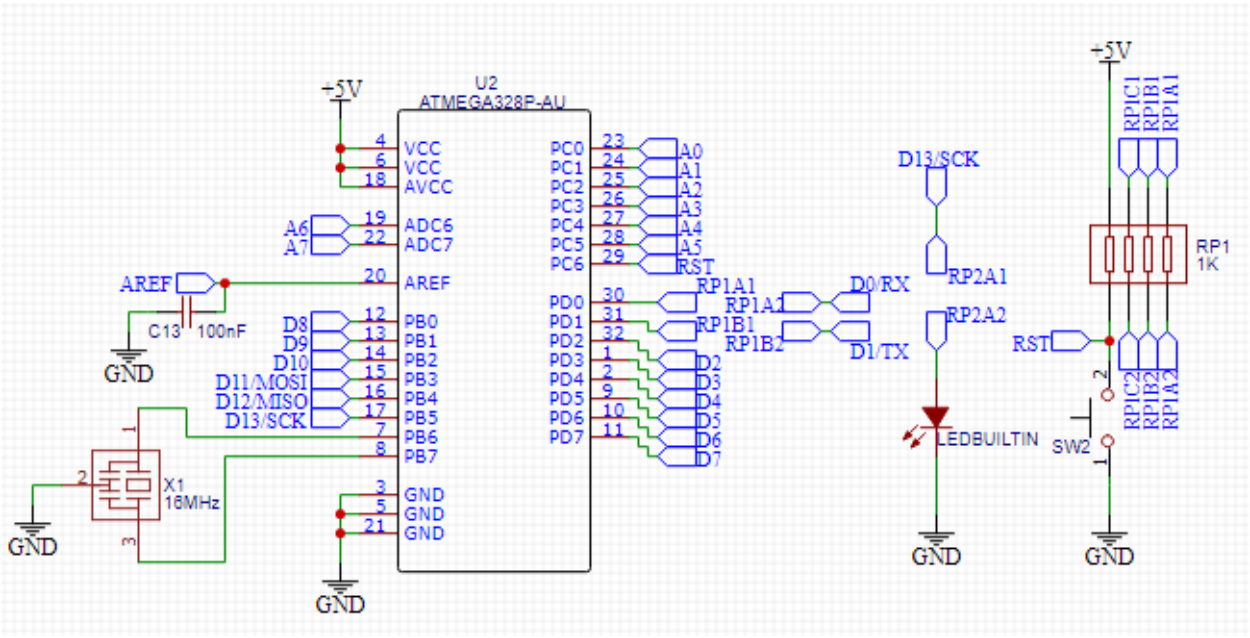


Рисунок 2.7 – Схема мікроконтролера з обв’язкою

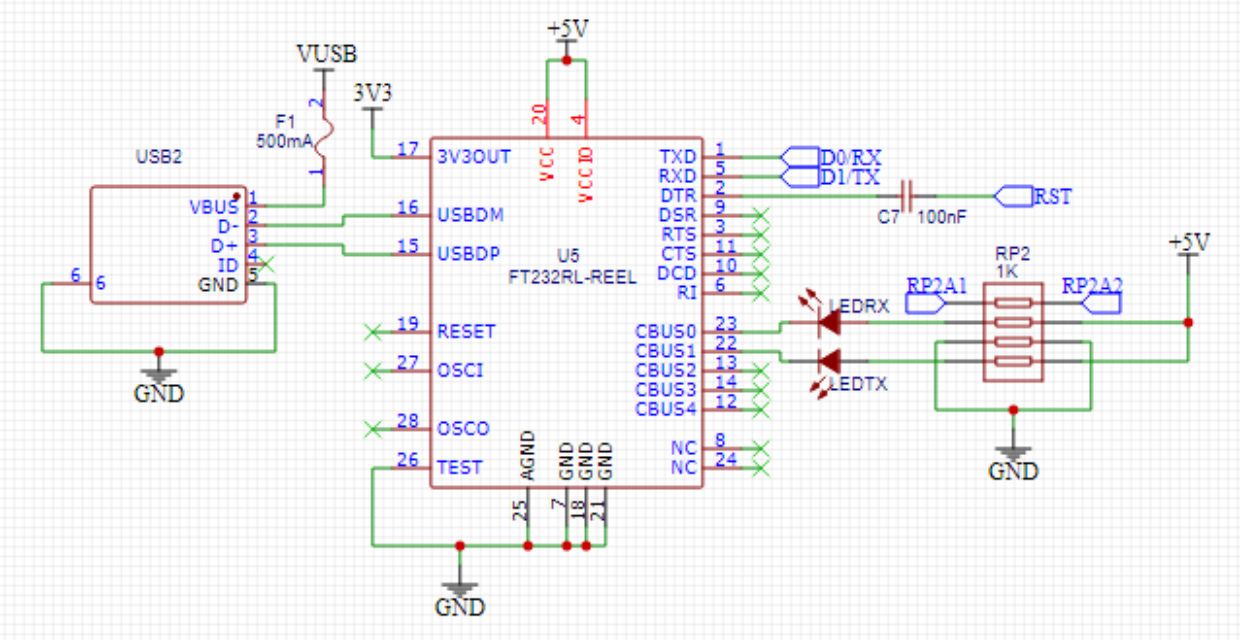


Рисунок 2.8 – Схема перетворювача TTL-USB

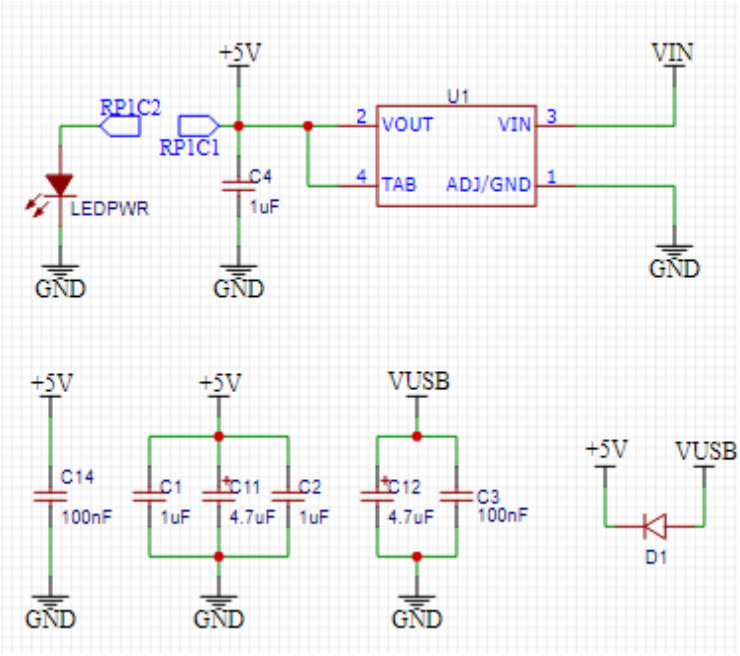


Рисунок 2.9 – Схеми живлення плати

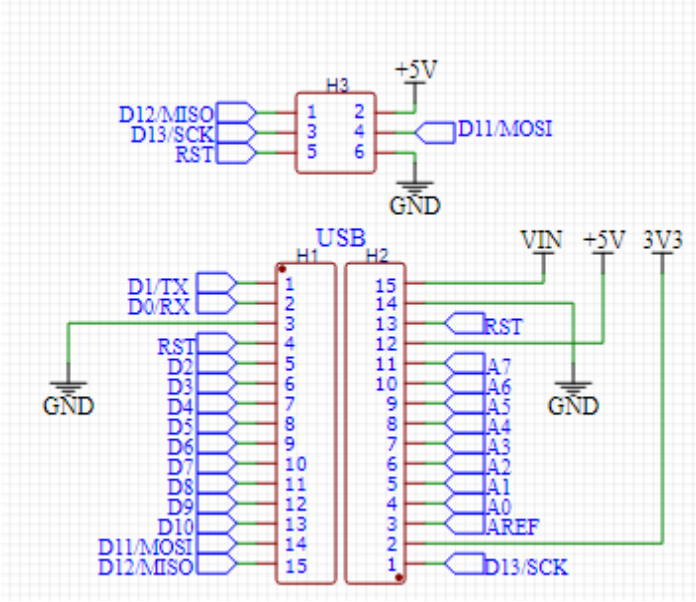


Рисунок 2.10 – Схеми комутації

На наступному етапі було створено схему RFID зчитувача RC522 згідно його документації (рис.2.11).

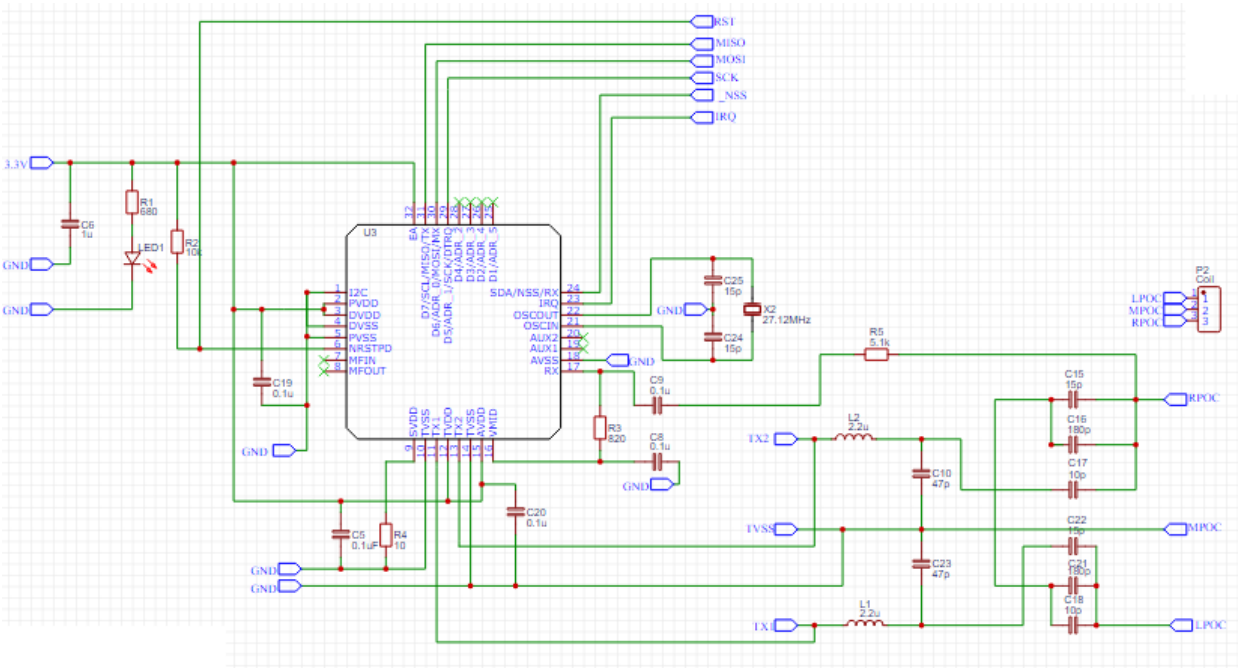


Рисунок 2.11 – Схема зчитувача

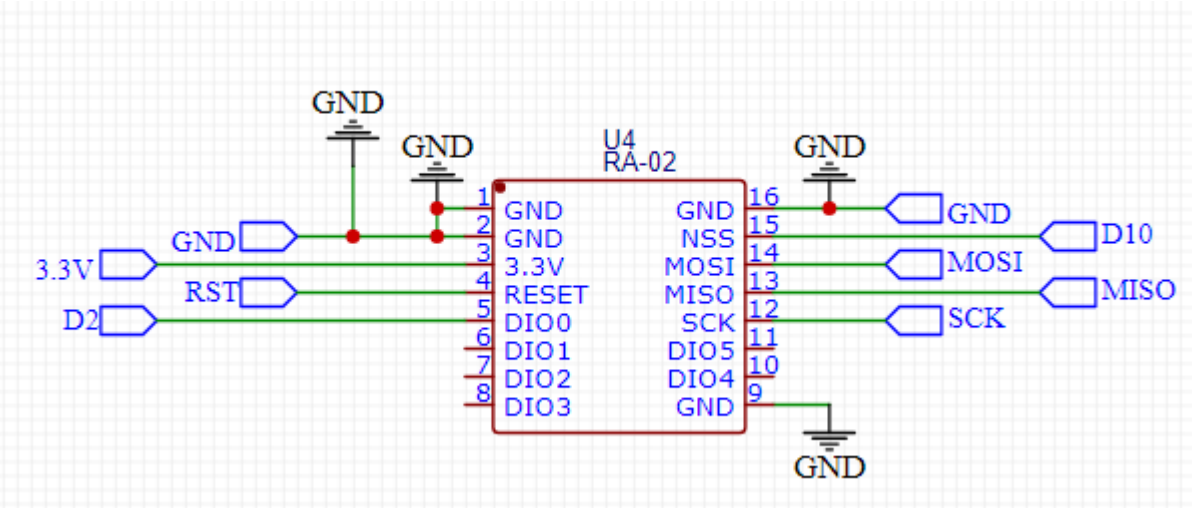


Рисунок 2.12 – Модуль прийомо-передавача LoRa SX1278

Для з’єднання модульних елементів між собою в програмі EasyEda відповідні контакти позначені між собою відповідними контактами.

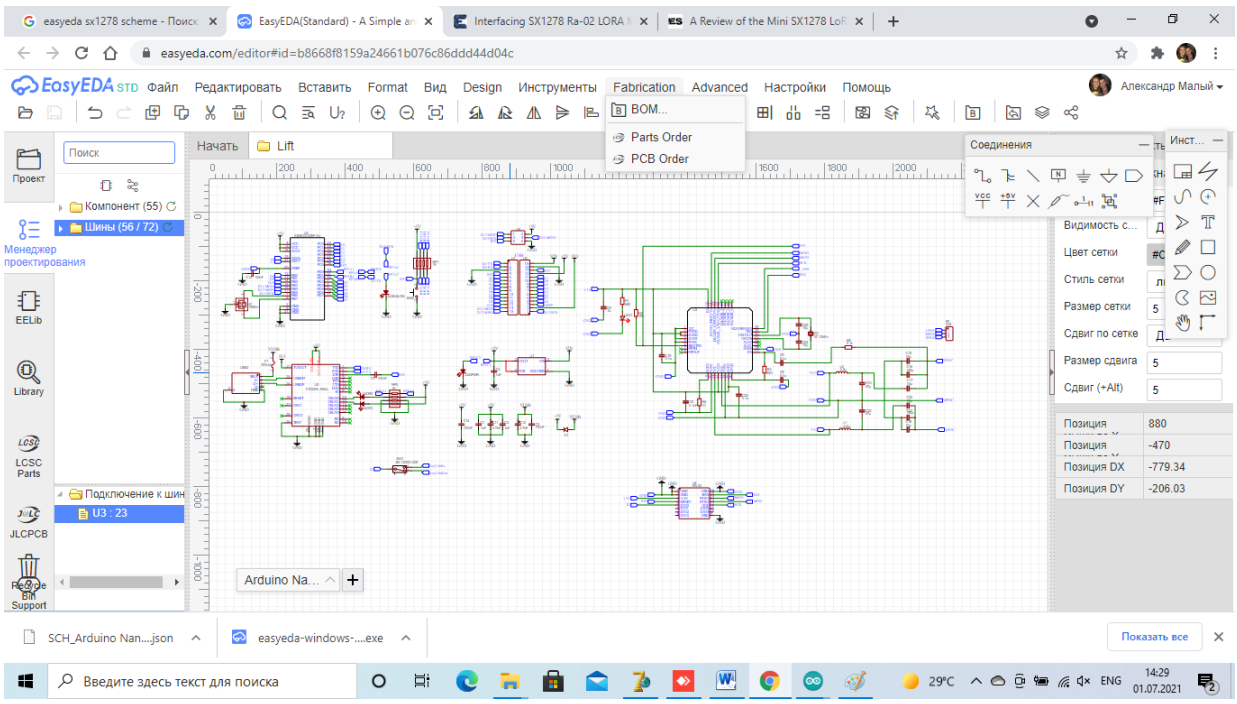


Рисунок 2.13 – Загальна схема модуля виконана в програмі EasyEda

Повна схема модулю наведена у додатку А.

2.3 Розробка конструкції модулю

Оскільки клавіатура ліфтів є стандартною та ми впливаємо лише замиканням/розмиканням дроту живлення нам немає необхідності розробки клавіатури ліфту, а треба лише визначити який саме контакт ми будемо перемикати для різних ліфтів

Модуль своїм гальванично-розв'язаним контактом (релейним модулем) розриває загальний ланцюг живлення клавіатури вибору поверху і підключає його тільки тоді, коли піднесена електронна безконтактна картка або брелок з дозволом на поїздки.

Для ліфтів 320кг - розрив 11-го проводу, 400кг (релейна станція) -разрив 187-го проводу, 500кг (кнопки з залипання) - розрив 279-го проводу + додатково в розрив кожної кнопки вибору поверху по одному діоду, УПЛ-розрив 3 го дроту, що йде до пульта вибору поверху. Для ліфтів серії ВУЛ

(мікропроцесор \ Могильов) - для кожної групи матриці: 1 - 8 поверхи - розрив 601-го проводу, 9 - 16 поверх - розрив 602-го проводу.

Релейний модуль та зчитувач обрано стандартні, а отже була необхідність розробки лише мікропроцесорного модуля керування та LoRa модулю приймання/передавання даних.

Розробка друкованих плат мікропроцесорного модуля та модулю приймання/передавання даних була зроблена у програмі Sprint Layuot (рис.2.14-2.15).

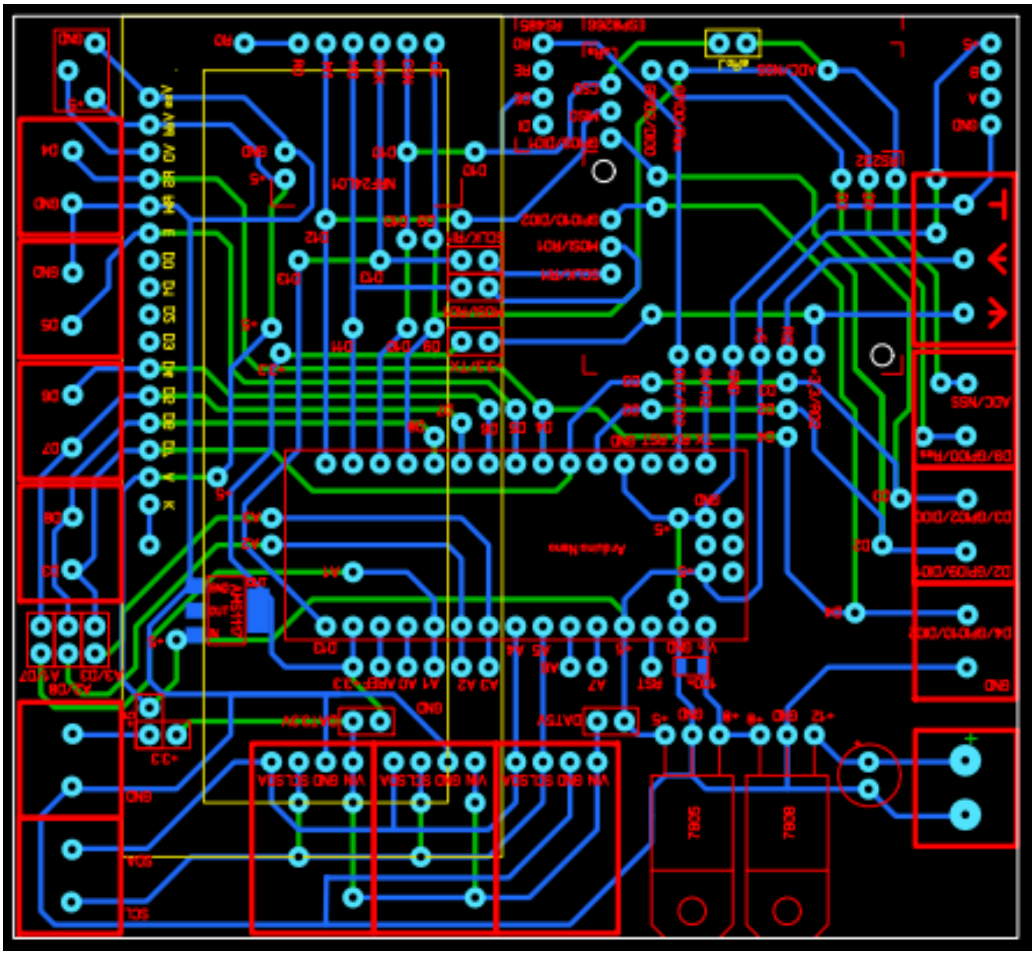


Рисунок 2.14 – Розроблена конструкція друкованої плати мікропроцесорного модуля керування в програмі Sprint Layout

На наведених рисунках в програмі Sprint Layout синім кольором виконуються друковані провідники сторони встановлення елементів, зеленим друковані провідники зворотної сторони (слой виконано таким чином як би

дивилися через прозору плату), блакитним кольором виконано металізовані отвори між слоями, червоним кольором виконано маркування сторони встановлення елементів.

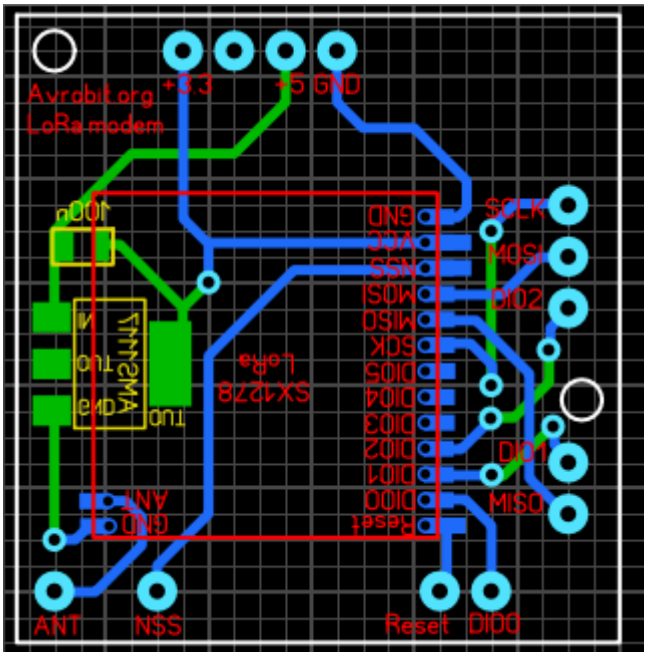


Рисунок 2.15 – Розроблена конструкція друкованої плати LoRa модулю приймання/передавання даних в програмі Sprint Layout

Плата приймання/передавання даних виконана таким чином, щоб вона з’єднувалась безпаячним монтажем з мікропроцесорним модулем керування за допомогою PLS штекерів (з боку плати приймання/передавання) та PLD роз’ємів (з боку плати керування). З’єднання елементів наведено на фото виготовленої конструкції (рис. 2.16).

Плата виконана на фольгованому двосторонньому стеклотекстоліті комбінованим методом. Виготовлення друкованої плати булу замовлено на онлайн сервісі PCBWAY в Китаї та отримано поштовою доставкою.

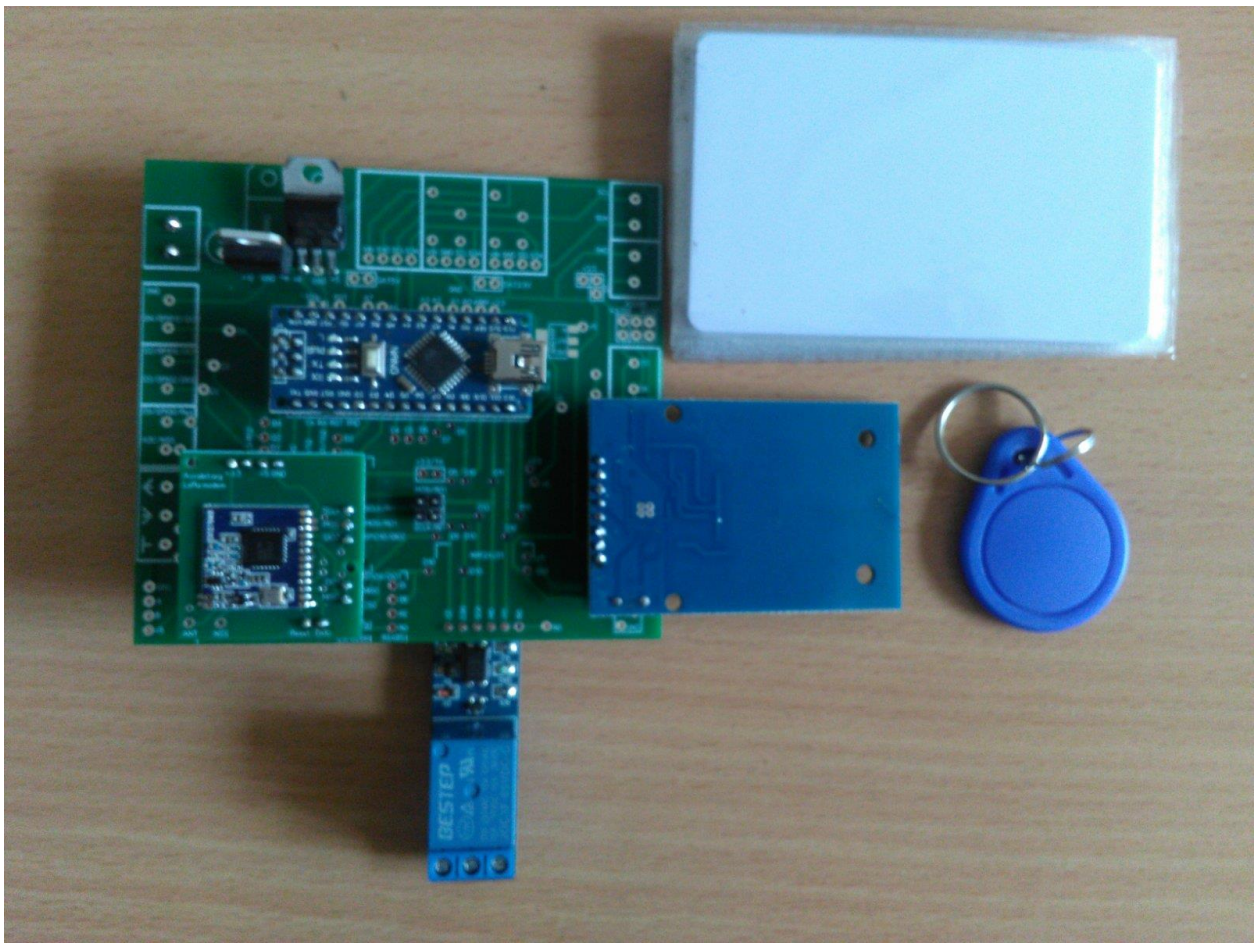


Рисунок 2.16 – Виготовлені друковані плати з розміщеними елементами

У якості корпуса для модуля було обрано базову несучу конструкцію – пластикову розподільчу коробку 8135КА (рис.2.17).

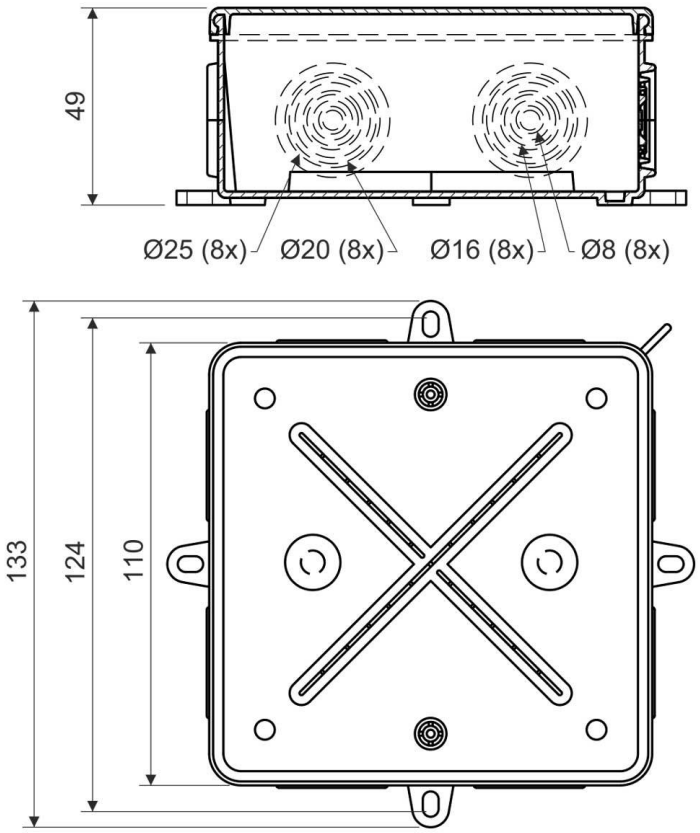


Рисунок 2.17 – Габаритні розміри корпусу

Для корпуса була розроблена 3Д модель в програмі Blender 2.93.1.

Розробка моделі включала наступні етапи:

- була створена площина, шляхом натискання кнопки “Додати” та вибору пункту “Меш”;

- за допомогою режиму редагування була створена базова форма короба розподільчої коробки;

- були створені меші два “Куб” та “Площина”, які будуть тримачами електроніки, кріпленнями (ніжками) та кришкою, відповідно;

- після створення базових форм в режимі редагування всіх трьох елементів, у тому же режимі було почато процес детальної проробки моделі (отвори, вирізи і т.д.);

- для оптимізації та прискорення процесу створення моделі були використані модифікатори “симетрія”, що віддзеркалює меші, та “Фаска”, що створює скоси та закруглення на кутах;

- після завершення деталізації було налаштовано Рендер шляхом вибору движка “Workbench”, налаштування камери та збільшення якості картинки.

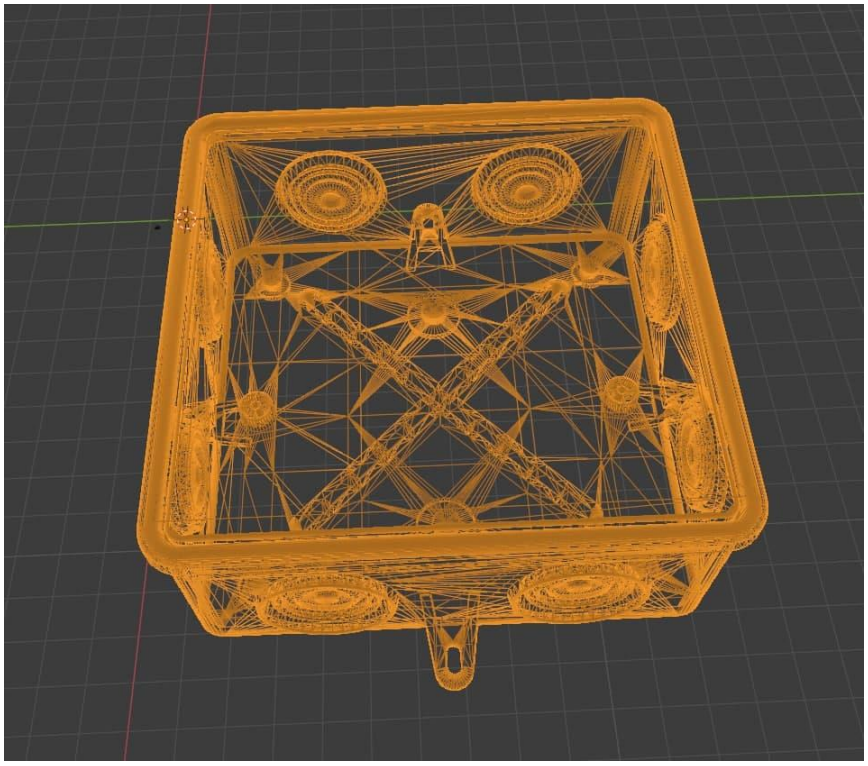


Рисунок 2.18 – Побудова моделі

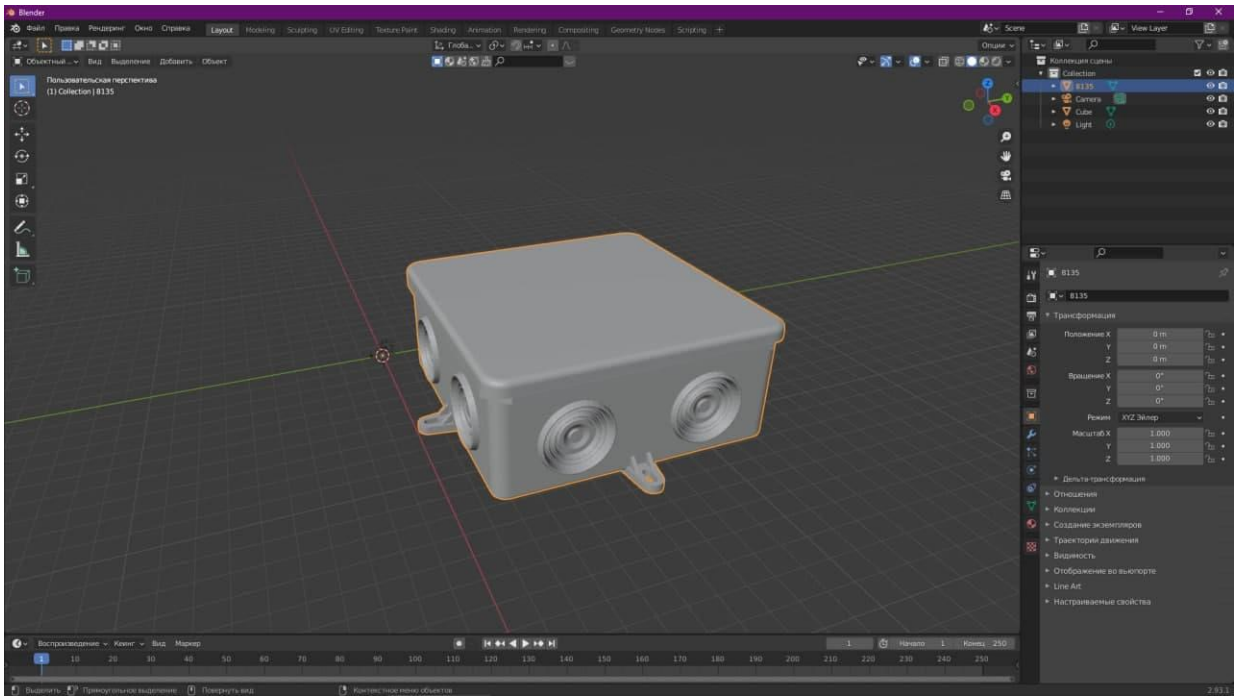


Рисунок 2.19 – Остаточна побудована модель в програмі Blender

Модуль, що встановлюється для контролю доступу для проїзду в ліфті з боку кабіни ліфта, монтується в кабіні праворуч від панелі з кнопками поверхів. Для цієї мети вирізається електролобзиком прямокутний отвір висотою - 85мм і шириною - 95мм збоку (або в будь-якому іншому зручному для монтажу і користування місці), після чого туди вставляється картоприймач і модуль контролю та закріплюється двома гвинтами. Виходить, передня панель картоприймача щільно прилягає до купе кабіни (аналогічно панелі кнопок вибору поверху). Габарити картоприймача: висота - 125мм, ширина - 100мм, глибина - 50мм. Кріплення передньої панелі здійснюється двома спецгвинтами з внутрішнього боку кабіни ліфта або шпильками з боку шахти (при необхідності додаткового кріплення на деяких матеріалах). Конструкція захисту зчитувача від вандального впливу наведена на рис.2.20.

Рисунок 2.20 – Антивандальна конструкція зчитувача, що встановлюється в кабіну ліфту

Пасажир заходить в ліфт, підносить картку або брелок до картоприймача, якщо модуль дозволив поїздку він через реле замикає попередньо розімкнутий їм ланцюг дроту живлення клавіатури вибору поверху та пасажир натискає кнопку потрібного поверху і ліфт починає рухатися попередньо закривши двері.

Алгоритм встановлення модулю контролю доступу в ліфти наведено на рис. 2.21.

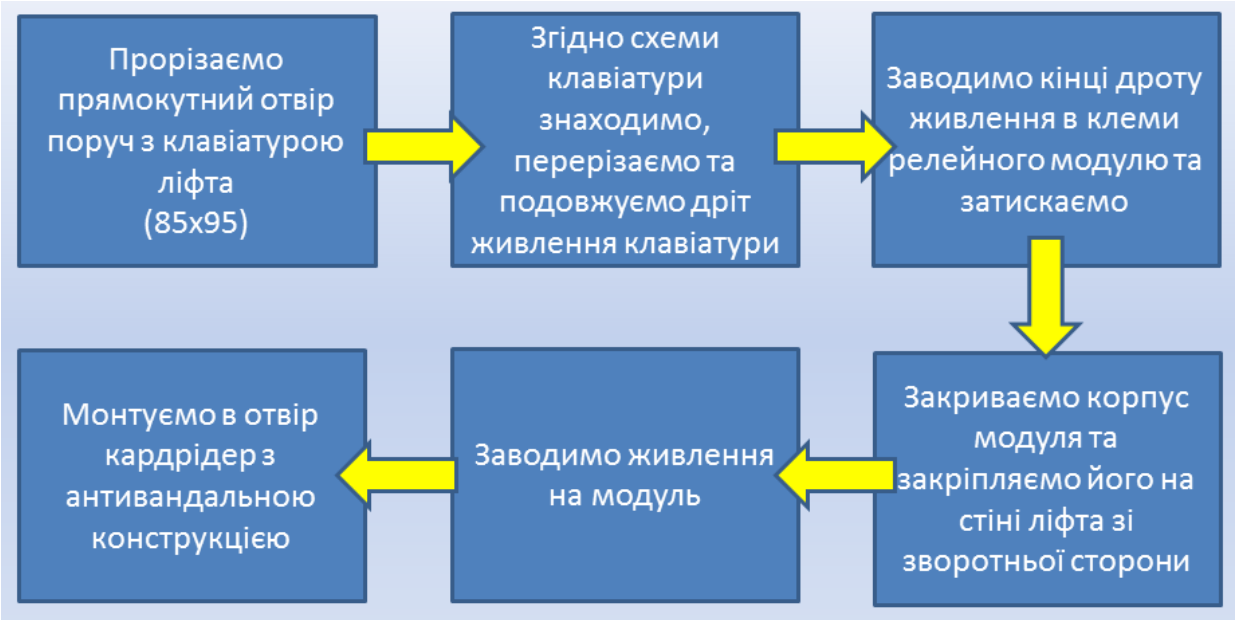


Рисунок 2.21 – Алгоритм встановлення модулю в ліфти

Переваги розробленої конструкції:

- швидкий монтаж у кабіну ліфту;

- не потребує додаткових дротових з'єднань;

- підключення до вузла збору по території і сервера проводиться автоматично по серійному номеру;

- перелік ключів та дозволів задається віддалено – немає потреби виїзду спеціаліста к ліфту

- можливість віддаленого контролю працеспроможності;

- модуль виконано на простих та недорогих елементах;

- не впливає на працездатність основної електроніки ліфту.

3 Написання програмного забезпечення

3.1 Розробка бази даних

Перед створенням програмного забезпечення для керування модулем доступу необхідно розробити структуру збереження даних та протокол обміну апаратної частини з віддаленим сервером.

Оптимальним з точки зору швидкості доступу є зберігання даних у базах даних. Було обрано SQL базу даних, оскільки такі БД мають можливості розширення та стандартизовані з точки зору зберігання та отримання даних, а також найчастіше використовуються на хостингах віддалених серверів.

Було розроблено структуру БД, що складається з 13 таблиць (табл.3.1-3.13).

Для зменшення об’єму текстової інформації БД було максимально проіндексовано та винесено дані, що можуть повторюватися у окремі таблиці. Розроблена база даних була створена реляційною оскільки дані основної таблиці (користувачів з їх параметрами та ключами доступу) пов’язані з повторюваними даними (квартир, домів, вулиць і т.і.). Робота з базою даних відбувається за допомогою SQL запитів.

Для скорочення кількості текстової інформації в основних таблицях, стосовно додаткового опису параметрів та нотаток, було створено таблицю описів (табл.3.1). Ця таблиця зберігає проіндексовані тексти, що можуть використовуватись для опису додаткових параметрів різних елементів структури БД. Створення цієї таблиці покликане насамперед не виділяти велике за розміром текстове поле в інших таблицях, а тільки у разі необхідності зробити посилання.

Стовбець Id є ключовим унікальним стовбцем, що покликаний ідентифікувати запис (інші таблиці посилаються саме на цій стовпець при індексації свої записів). Стовпець DescrText має на меті зберігати текст додаткового опису.

Таблиця 3.1 – Додаткові текстові описи Descriptions

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	DescrText	Текст опису	TEXT	–

SQL запит створення таблиці:

CREATE TABLE Descriptions (Id INT, DescrText TEXT, PRIMARY KEY (Id));

Оскільки розроблена система має у майбутньому встановлюватись на велику кількість ліфтів у різних населених пунктах було створено таблицю населених пунктів (табл.3.2) на яку будуть посилатися наступні таблиці з метою більш повного встановлення місцезнаходження. В одному населеному пункті може бути розташовано тисячі систем керування доступом в ліфтах, а отже назва населеного пункту буде повторюватись – зручніше це про індексувати та зробити у вигляді посилання.

Таблиця 3.2 – Населені пункти Cities

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	CityName	Назва населеного пункту	VARCHAR(100)	–
3	CityType	Тип населеного пункту	INT	–
4	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

Стовпець CityName зберігає назву населеного пункту у текстовому вигляді.

Стовпець CityType зберігає тип населеного пункту, який може приймати наступні значення:

- 0 (NULL) – тип населеного пункту не встановлено;
- 1 – місто;
- 2 – селище міського типу;
- 3 – село.

Стовпець Descr є посилання на додатковий опис населеного пункту у разі такої необхідності (наприклад наводити якість статистичні або історичні данні і т.і.).

SQL запит створення таблиці:

CREATE TABLE Cities (Id INT, CityName VARCHAR(100), CityType INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Для зберігання даних, щодо вулиць було створено окрему таблицю Streets (табл.3.3).

Таблиця 3.3 – Вулиці Streets

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	StreetName	Назва вулиці	VARCHAR(100)	–
3	StreetType	Тип вулиці	INT	–
4	CityId	Посилання на населений пункт	INT	Cities(Id)
5	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

Стовпець StreetName зберігає назву вулиці.

Стовпець StreetType зберігає тип вулиці та може приймати наступні значення:

- 0 (NULL) тип вулиці не встановлено;
- 1 – проспект;
- 2 – вулиця;
- 3 – бульвар;
- 4 – провулок;
- 5 – проїзд.

Стовпець CityId зберігає посилання на таблицю населених пунктів оскільки назви вулиць у різних населених пунктах можуть співпадати, а отже треба ідентифікувати яка саме вулиця мається на увазі.

SQL запит створення таблиці: CREATE TABLE Streets (Id INT, StreetName VARCHAR(100), StreetType INT, CityId INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (CityId) REFERENCES Cities(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Для ідентифікації окремих будинків, де встановлено систему керування доступом до ліфтів було створено таблицю будинків (табл.3.4).

Таблиця 3.4 – Будинки Houses

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	HouseNumb	Номер дому	VARCHAR(10)	–
3	StreetId	Посилання на вулицю	INT	Streets(Id)
4	Latitude	Широта	FLOAT	–
5	Longitude	Довгота	FLOAT	–
6	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

Стовпець HouseNumb зберігає номер будинку. Тип записів у цьому стовбці текстовий оскільки деякі будинки у номері мають букви, а отже такі значення не можна зберігати лише у числовому вигляді.

Стовпець StreetId зберігає індексне посилання на таблицю вулиць, щоб ідентифікувати до якої вулиці належить дім.

Стовбці Latitude та Longitude зберігають координати розташування будинку, що дозволяє робити пошук будинку на території (наприклад по Google карті) та виводити на карті його розташування. Це спрощує роботу з системою особливо з боку адміністраторів системи та представників керуючих компаній у яких кількість домів може сягати десятків та навіть тисяч (у випадку адміністраторів системи).

SQL запит створення таблиці:

CREATE TABLE Houses (Id INT, HouseNumb VARCHAR(10), StreetId INT, Latitude FLOAT, Longitude FLOAT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (StreetId) REFERENCES Streets(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Однією з основних таблиць системи керування доступом до ліфтів є таблиця квартир (табл.3.5) оскільки саме сплата за обслуговання будинку з боку власників квартир визначає право на проїзд у ліфті.

Таблиця 3.5 – Квартири Flats

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	FlatNumb	Номер квартири	VARCHAR(10)	–
3	HouseId	Посилання на будинок	INT	Houses(Id)
4	Porch	Номер під’їзду	INT	–
5	AccStatus	Статус доступу	INT	–
6	Descr	Посилання на дод. опис	INT	Descriptions(Id)

Кожна квартира має свій номер (як показав аналіз бази квартир Запоріжжя номер може бути не лише числовий), прив’язана до окремого будинку за допомогою посилання на таблицю будинків HouseId та знаходиться у якомусь під’їзді Porch (цей параметр важливий тому, що ліфт встановлюється на окремо на кожен під’їзд).

Для зберігання статусу доступу використовується стовпець AccStatus, що вказує чи можуть жильці цієї квартири користуватись ліфтом чи ні.

SQL запит створення таблиці:

CREATE TABLE Flats (Id INT, FlatNumb VARCHAR(10), HouseId INT, AccStatus INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (HouseId) REFERENCES Houses(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Оскільки часто прізвища, імена та по батькові у людей періодично збігаються було прийнято рішення створити окремі таблиці, що будуть зберігати текстові написи цих імен, що дозволить проводити пошук жителів більш швидко за рахунок індексації. Тобто для пошуку наприклад всіх Іванових достатньо знайти індекс цього прізвища та шукати в таблиці жильців числове значення, а не аналізувати текстове, а кількість безпосередньо жильців, навіть в межах населеного пункту, може сягати сотен тисяч.

Таблиця 3.6 – Прізвища Surnames

№	Назва стовбцю	Що зберігається	Тип даних
1	Id	Номер запису (ключ таблиці)	INT
2	Name	Текстовий напис прізвища	VARCHAR(100)

SQL запит створення таблиці:

CREATE TABLE Surnames (Id INT, Name VARCHAR(100), PRIMARY KEY (Id));

Таблиця 3.7 – Імена Names

№	Назва стовбцю	Що зберігається	Тип даних
1	Id	Номер запису (ключ таблиці)	INT
2	Name	Текстовий напис ім’я	VARCHAR(50)

SQL запит створення таблиці:

CREATE TABLE Names (Id INT, Name VARCHAR(50), PRIMARY KEY (Id));

Таблиця 3.8 – По батькові Fnames

№	Назва стовбцю	Що зберігається	Тип даних
1	Id	Номер запису (ключ таблиці)	INT
2	Name	Текстовий напис по батькові	VARCHAR(50)

SQL запит створення таблиці:

CREATE TABLE Fnames (Id INT, Name VARCHAR(50), PRIMARY KEY (Id));

Стовбці Name таблиць (3.6-3.8) зберігають текстовий напис у іменному відмінку відповідно прізвища, ім’я та по батькові.

Для зберігання даних про окремого жильця було створено таблицю Man (табл.3.9).

Стовбці SurNameId, NameId, FnameId зберігають посилання на текстове написання прізвища, ім’я та по батькові жильця.

Стовбець FlatId зберігає посилання на квартиру де проживає житель у таблиці квартир.

Стовбець Card зберігає унікальний ідентифікатор ключа доступу (картки, брелока або NFC на смарфоні).

Таблиця 3.9 – Жильці Man

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	SurNameId	Посилання на прізвище	INT	Surnames(Id)
3	NameId	Посилання на ім’я	INT	Names(Id)
4	FnameId	Посилання на по батькові	INT	Fnames(Id)
5	FlatId	Посилання на квартиру	INT	Flats(Id)
6	Card	Ідентифікаційний номер карти	CHAR(8)	
7	Phone	Мобільний номер телефону	VARCHAR(15)	
8	TelegramId	ІД у Telegram	INT	
9	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

Стовбець Phone зберігає мобільний номер телефону жильця для проведення двофакторної авторизації при вході в систему.

Стовбець TelegramId зберігає ІД користувача у меседжері Telegram, якщо жилець використовує цей месседжер та встановив собі бота помічника по системі.

SQL запит створення таблиці:

CREATE TABLE Man (Id INT, SurNameId INT, NameId INT, FNameId INT, FlatId INT, Card CHAR(8), Phone VARCHAR(15), TelegramId INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (SurNameId) REFERENCES Surnames(Id), FOREIGN KEY (NameId) REFERENCES Names(Id), FOREIGN

KEY (FNameId) REFERENCES FNames(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Для зберігання інформації, щодо ліфтів в які встановлено обладнання контролю доступу розроблено таблицю Elevators (табл.3.10).

Ця таблиця вказує в якому домі встановлена система HouseId, дату встановлення EquipDate, унікальний ідентифікатор апаратного модулю контролю, що встановлений у цей ліфт EquipID, тип лифтового обладнання (модель ліфту), номер під'їзду у домі де встановлено обладнання Porch.

Таблиця 3.10 – Ліфти Elevators

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	EquipDate	Дата встановлення обладнання контролю доступу	DATE	–
3	EquipID	Унікальний ІД модулю контролю встановлений у ліфті	VARCHAR(20)	
4	Type	Тип ліфту	INT	–
5	HouseId	Посилання на будинок	INT	Houses(Id)
6	Porch	Номер під'їзду	INT	
7	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

SQL запит створення таблиці:

CREATE TABLE Elevators (Id INT, EquipDate DATE, EquipID VARCHAR(20), Type INT, HouseId INT, Porch INT, Descr INT , PRIMARY KEY

(Id), FOREIGN KEY (HouseId) REFERENCES Houses(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Для зберігання даних щодо керуючих компаній та ОСББ було розроблено таблицю Companies (табл.3.11), що зберігає назву компанії Name, її тип (керуюча компанія або ОСББ) Type та у разі необхідності додатковий опис як посилання на таблицю описів.

Таблиця 3.11 – Керуючі компанії Companies

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	Name	Назва компанії	VARCHAR(200)	–
3	Type	Тип компанії	INT	–
4	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

SQL запит створення таблиці:

CREATE TABLE Companies (Id INT, Name VARCHAR(200), Type INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Оскільки одна керуюча компанія, ОСББ або асоціація власників як правило мають у обслугованні більш одного будинку та відповідно буде встановлено не одну систему контролю доступу до ліфтів було створено таблицю відповідності будинків керуючим компаніям. Ця таблиця використовується для відображення переліку домів та ліфтів з системою керування при виборі окремої компанії та/або вході вповноваженої особи для перегляду та редагування даних.

Таблиця 3.12 – Відповідність будинків керуючим компаніям HouseOwners

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	–
2	CompanyId	Посилання на керуючу компанію	INT	Companies(Id)
3	HouseId	Посилання на будинок	INT	Houses(Id)
4	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

SQL запит створення таблиці:

CREATE TABLE HouseOwners (Id INT, CompanyId INT, HouseId INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (CompanyId) REFERENCES Companies(Id), FOREIGN KEY (HouseId) REFERENCES Houses(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Для керування входами користувачів у систему створено таблицю Users яка покликана зберігати дані авторизації користувачів системи при вході через термінальне ПЗ та WEB сторінку користувача/адміністратора.

Таблиця зберігає логін користувача Login, хеш паролю PassHash, посилання безпосередньо на людину (адже навіть адміністратор системи десь проживає) ManId, рівень доступу AccLvl, посилання на керуючу компанію, що обслуговує квартиру, в якій проживає жилець (у випадку користувача) або до адміністрування якої відноситься адміністратор CompanyId, посилання на квартиру проживання у випадку користувача FlatId (для адміністратора це поле буде мати значення NULL).

Таблиця 3.13 – Користувачі Users

№	Назва стовбцю	Що зберігається	Тип даних	На які таблиці посилається
1	Id	Номер запису (ключ таблиці)	INT	
2	Login	Логін користувача	VARCHAR(100)	
3	PassHash	Хеш паролю	CHAR(32)	
4	ManId	Посилання на людину	INT	Man(Id)
5	AccLvl	Рівень доступу	INT	
6	CompanyId	Посилання на керуючу компанію	INT	Companies(Id)
7	FlatId	Посилання на квартиру проживання	INT	Flats(Id)
8	Descr	Посилання на додатковий опис	INT	Descriptions(Id)

Рівень доступу може приймати наступні значення:

- 0 (NULL) – звичайний користувач якому доступний лише перегляд поточного стану доступу до квартири де він проживає та перелік жильців цієї квартири з ІД карток доступу;

- 1 – уповноважена особа керуючої компанії (ОСББ, асоціації власників), що має можливість перегляду стану доступів всіх квартир всіх домів, що обслуговуються компанією;

- 2 – уповноважена особа керуючої компанії (ОСББ, асоціації власників), що має можливість редагування стану доступів, додавання та видалення ключів доступу, редагування, додавання та зміни жильців всіх квартир всіх домів, що обслуговуються компанією;

- 3 – адміністратор системи, що має можливості перегляду, редагування всієї бази системи.

SQL запит створення таблиці:

CREATE TABLE Users (Id INT, Login VARCHAR(100), PassHash CHAR(32), ManId INT, AccLvl INT, CompanyId INT, FlatId INT, Descr INT , PRIMARY KEY (Id), FOREIGN KEY (ManId) REFERENCES Man(Id), FOREIGN KEY (CompanyId) REFERENCES Companies(Id), FOREIGN KEY (FlatId) REFERENCES Flats(Id), FOREIGN KEY (Descr) REFERENCES Descriptions(Id));

Базу даних було створено та розміщено на віддаленому сервері хостингера ukraine.com.ua з підтримкою СУБД MySQL. Для створення БД в панелі керування хостингу було обрано пункт «Бази даних» (рис.3.1).

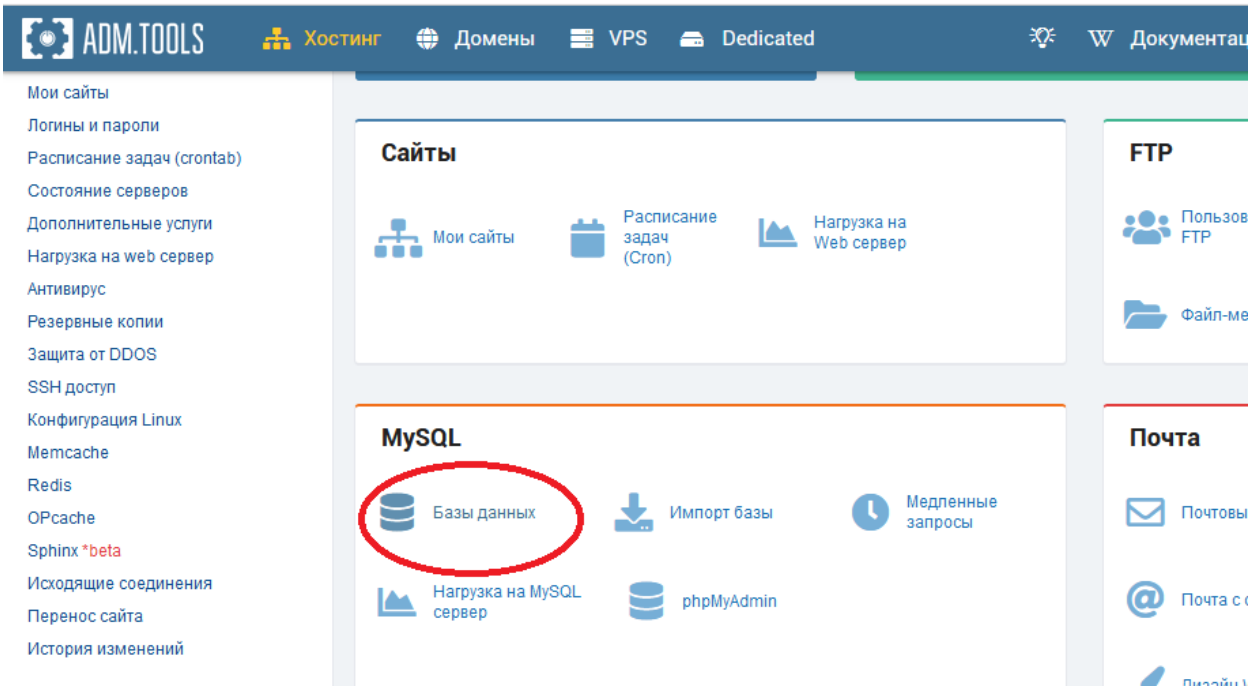


Рисунок 3.1 – Панель керування базами даних хостингу

Далі для створення самої бази було обрано пункт «Створити базу даних» та введено ім’я «lifts» (рис.3.2).

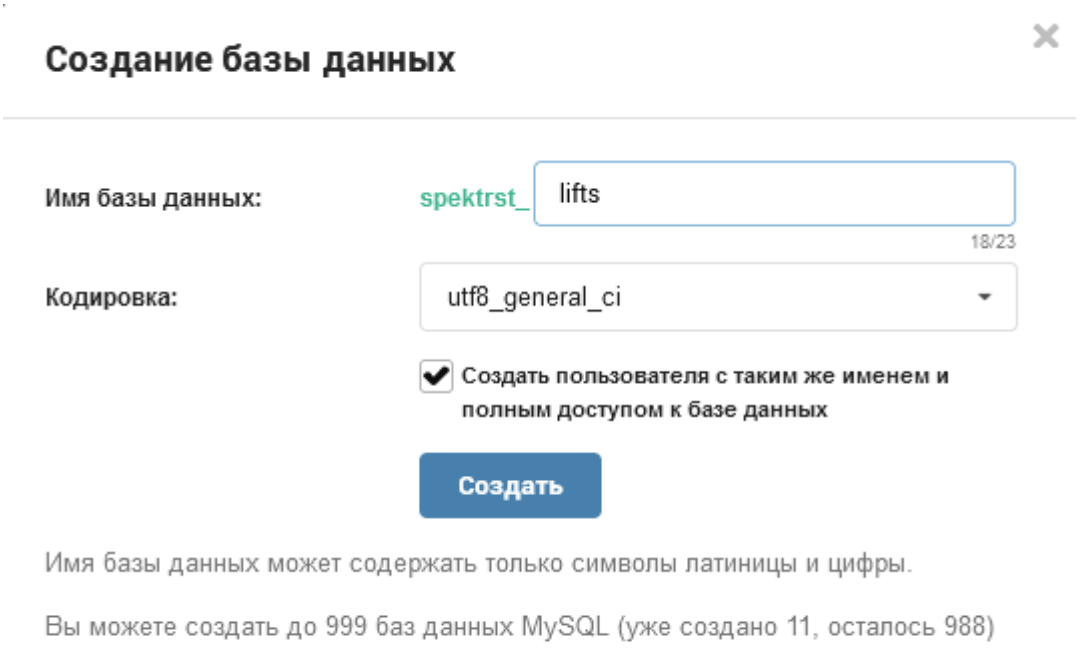


Рисунок 3.2 – Створення бази даних з ім’ям «lifts»

Оскільки заздалегідь було розроблено структуру кращим варіантом створення таблиць є запуск панелі керування SQL запитами phpMyAdmin та запуск запитів у відповідному полі (рис.3.3).

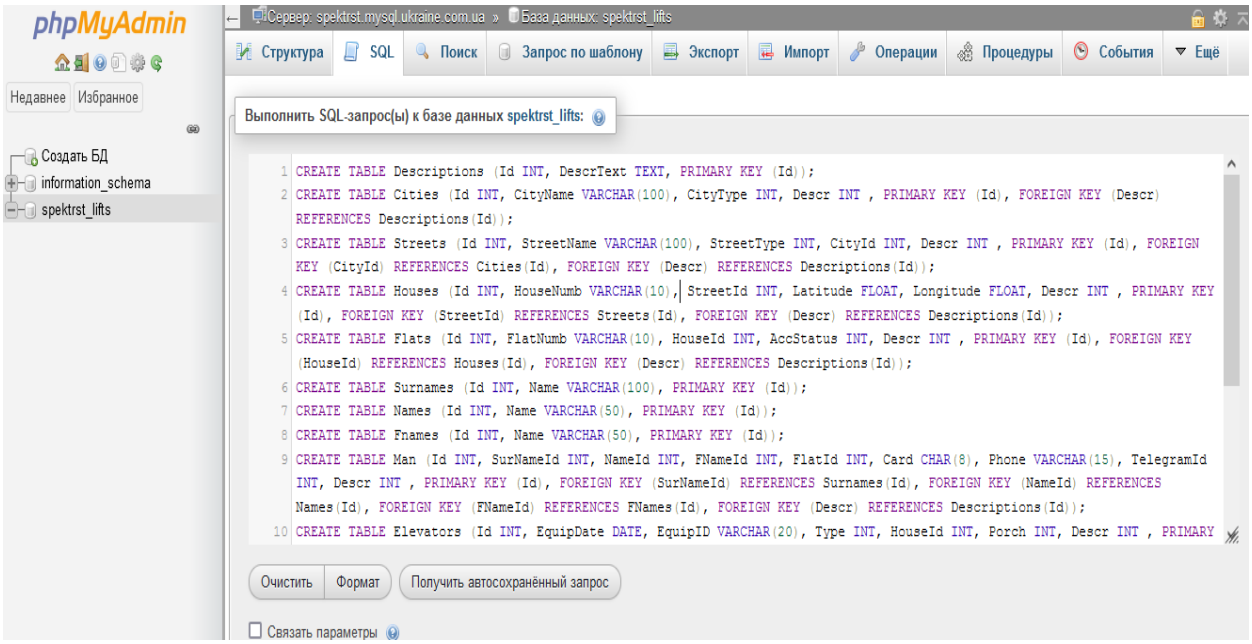


Рисунок 3.3 – Виконання SQL запитів створення БД в phpMyAdmin

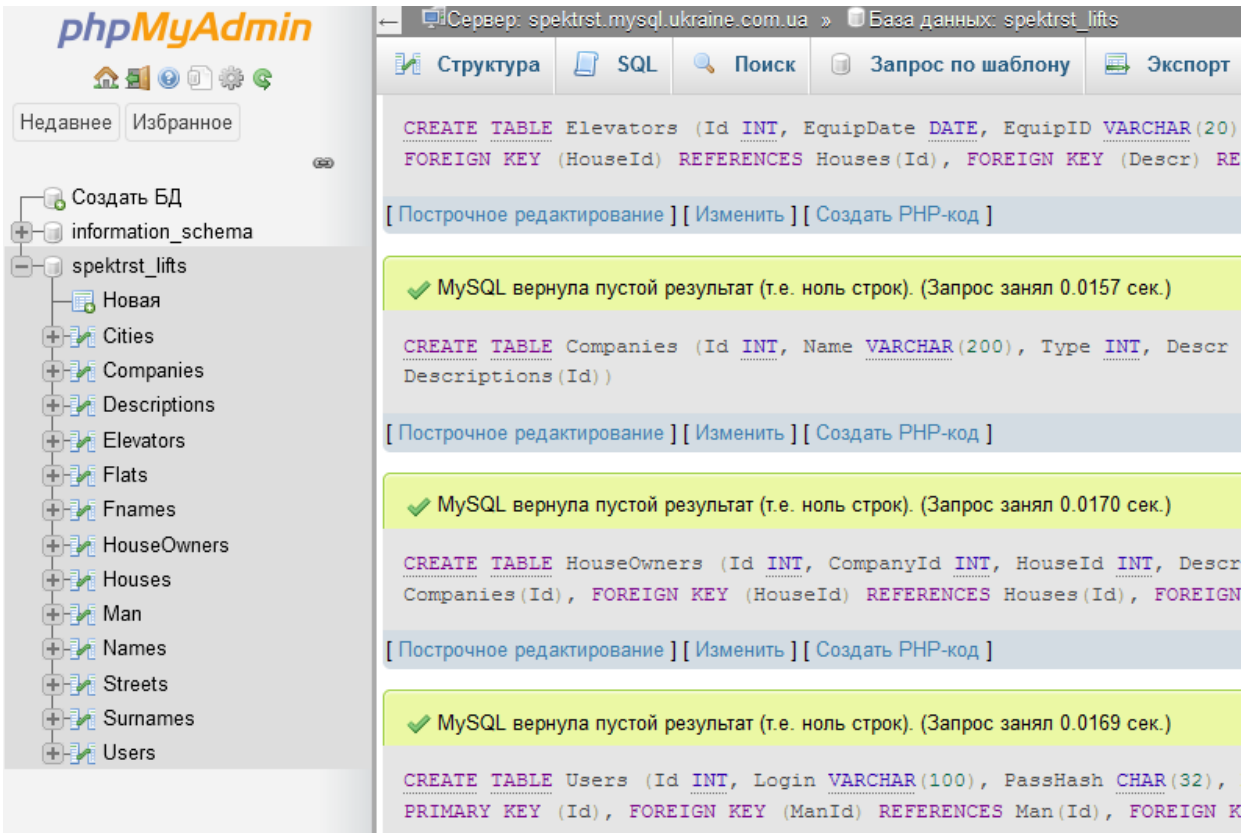


Рисунок 3.4 – Створена БД на хостунгу

3.2 Протоколи обміну між апаратним забезпеченням системи

Після створення структури БД та розробки апаратної частини модулю контролю доступу для написання керуючої програми мікроконтролера було проведено розробку протоколу обміну даними між окремими модулями керування доступом та вузлом збору даних на території, а також між вузлом збору даних та віддаленим сервером.

У табл.3.14 наведено команди обміну даними між модулем керування доступом, що встановлюються безпосередньо у ліфті та вузлом збору даних на території.

Для команд 3 та 4 модуль дублює отриману від вузла інформацію, щоб вузол був впевнений, що інформація була коректно отримана модулем. Команда 5 ініціюється вузлом у випадку коли модулю треба почекати з обміном даними.

Оскільки для передавання даних використовується бездротова мережа LoRa, що має низьку швидкість команди були створені невеликої довжини для прискорення обміну необхідною інформацією.

Загалом модулю керування достатньо оновлювати перелік ключів, що прив’язані до модулю та їх стан щодо доступу. Модуль раз в годину з’єднується з вузлом збору даних на території та отримує оновлений список ключів та їх дозволів та при виявленні змін вносить їх у свою пам'ять для подальшої роботи.

Таблиця 3.14 – Команди обміну між модулем керування доступом та вузлом збору даних на території

№	Пакет з боку модуля	Пакет з боку вузла	Опис
1	02 ID ID ID ID 81 81 03	02 ID ID ID ID FF 03 або 02 ID ID ID ID 00 03	Запросити чи була зміна у переліку та статусі ключів доступу ID ID ID ID – унікальний ідентифікаційний номер апаратного модулю доступу, FF зміни були, 00 змін не було)
2	02 ID ID ID ID 0F 81 03		Підтвердження отримання модулем інформації щодо статусу змін
3	02 ID ID ID ID 55 AA 03	02 ID ID ID ID NN NN 03	Отримати кількість ключів доступу (NN NN – 2 байти у шістнадцятковій системі числення з кількістю ключів доступу, що приписані до модуля з яким проводиться обмін даними).

4	02 ID ID ID ID FF 00 NN NN 03	02 ID ID ID ID NN NN KK KK KK KK XX 03	Отримати Nй ключ доступу зі статусом (NN NN – 2 байти у шістнадцятьковій системі числення з номером ключа статус та ІД якого треба оновити ключів доступу, що приписані до модуля з яким проводиться обмін даними
5	02 ID ID ID ID C3 3C 03	02 ID ID ID ID C3 3C 03	Перейти в режим очікування на 1 хвилину (ініціюється вузлом збору на території у разі якщо в даний момент відбувається пересилання даних іншому модулю)
6	02 AA ID ID ID ID 03	02 AA ID ID ID ID 03	Перевірка доступності модуля з боку вузла

Час зв’язку модулів з вузлом збору даних рознесено таким чином, щоб виключити одночасний запит з декількох модулів до вузлу. Наприклад модуль 1го під’їзду 12го дому буде зв’язуватися у 5 хвилин від початку години, а модуль 2го під’їзду того ж дому у 7 хвилин від початку години. Додатково модулі рознесені по різних каналах та у разі одночасного запиту рангуються по пріоритету вузлом збору для чого існує спеціальна команда.

Якщо під час обміну інформацією один з елементів передачі (модуль або вузол) не отримав даних у визначеному форматі він повторює свій запит.

Оскільки вузол збирання даних по території зв’язується з сервером за рахунок більш швидкісного з’єднання (за допомогою бездротових мереж WiFi або GPRS , або за рахунок дротового з’єднання Ethernet з маршрутизатором приєднаним до мережі інтернет) обмін даними між ним та сервером

відбувається за допомогою POST запитів з використанням JSON для спрощення передачі масивів даних.

Запит з боку вузла до сервера на отримання масиву ідентифікаторів карток для апаратного модуля керування доступів має наступний вигляд:

{

Command:”GETCards”,

NodeID:<NodeID>,

NodeKey: <NodeKey>,

ModuleID: <ModuleID>

}

де параметр command характеризує команду запиту;

- параметр NodeID вказує унікальний ідентифікатор вузла;

- параметр NodeKey вказує ключ доступу, що має відповідати ідентифікатору вузла;

- параметр ModuleID вказує унікальний ідентифікатор модуля для якого треба отримати масив ключів доступу з їх статусом.

У відповідь на запит вузла сервер відправляє JSON масив наступної структури:

{

NodeID:<NodeID>,

ModuleID:<ModuleID>

Keys:[

{

KeyID:<KeyID>,

KeyStatus:<KeyStatus>

},{…},{…}]

}

де параметр NodeID вказує унікальний ідентифікатор вузла для якого висилається масив ключів;

- параметр ModuleID вказує унікальний ідентифікатор модуля для якого відправляється перелік ключів;

- масив Keys безпосередньо включає перелік ключів зі статусами доступу;

- параметр KeyID – унікальний ідентифікатор ключа доступу;

- параметр KeyStatus характеризує статус доступу ключа (0 якщо доступ заборонено та 1 якщо доступ дозволено).

При поточній реалізації не передбачено перегляд списку кількості та часу поїздок для кожного окремого ключа, однак при подальшій модернізації системи можливо передавати ці параметри.

3.3 Написання коду програми модуля керування доступом

Однією з основних частин при написанні коду програми модуля керування доступом є програмування обміну даним через модуль LoRa.

LoRa SX1278 підключається виводом VCC для живлення до 3,3 В Arduino. Також усі виводи GND модему підключаються до GND Arduino. Вивід RST підключено до D9, а вивід DIO0 до D2 Arduino. Виводи SPI NSS, MOSI, MISO, SCK підключені до відповідних виводів Arduino D10, D11, D12, D13 які призначені для обміну даними з пристроями з шиною SPI.

Для зв'язку між двома модулями LoRa нам потрібно мати бібліотеку LoRa. Щоб отримати LoRa Library, переходимо до менеджера бібліотеки та знаходимо LoRa та встановлюємо її (рис.3.1).

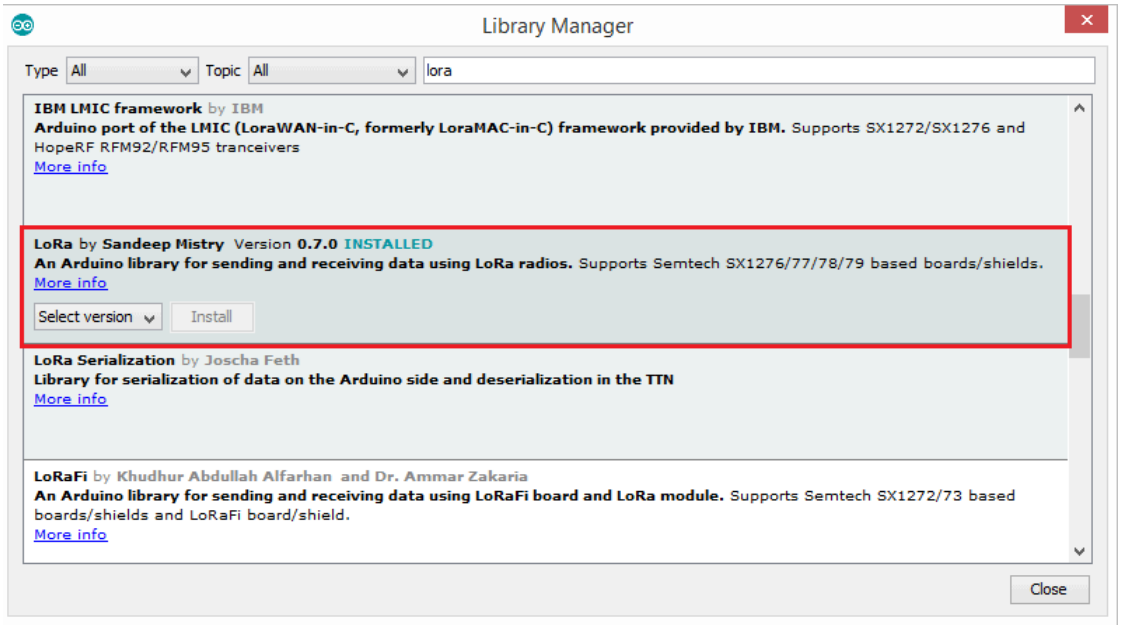


Рисунок 3.1 – Встановлення бібліотеки для роботи з модулем LoRa

Для роботи бібліотеки підключаємо її до коду програми #include <LoRa.h>

Після підключення бібліотеки можемо використовувати та перевіряти коди по роботі з модулем передавання LoRa.

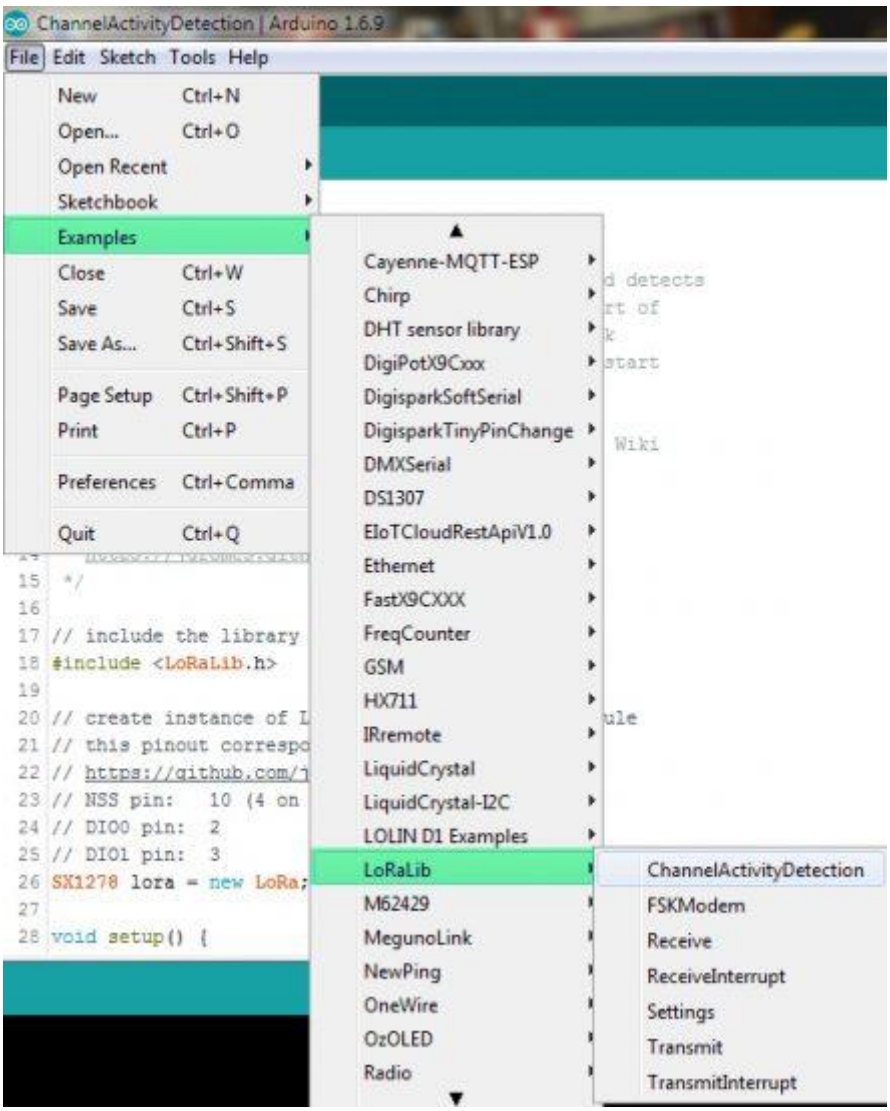


Рисунок 3.2 – Блоки програм для тестування модулів LoRa

Для перевірки зв’язку між модулями LoRa було написано код передавача та код приймача. Статус модулів передається у серійний порт через USB з’єднання кабелем через роз’їм, що встановлено на самій платі Arduino.

Код передавача:

Підключення бібліотек:

#include <SPI.h> – бібліотека для роботи з пристроями по SPI

#include <LoRa.h> – бібліотека для роботи з модулями LoRa

Опис виводу до якого приєднано змінний резистор , що буде передавати дані по LoRa модему (для тестування можемо змінювати опір резистору та значення буде передаватися приймачу):

int pot = A0;

Блок первинних налаштувань:

```
void setup() {
    Serial.begin(9600);//запуск серійного порту для передавання статусу передавача через USB
    pinMode(pot,INPUT); //вказання напрямку роботи виводу до якого підключено резистор, що зчитується для тестування передавання
    while (!Serial);
    Serial.println("LoRa Transmitter");//повідомлення в порт, що передавач запущено
    if (!LoRa.begin(433E6)) { // або 915E6, швидкість у МГц для модуля
        Serial.println("Start LoRa error!"); //повідомлення про помилку з’єднання
        while (1);
    }
}
```

Основний блок перевірки роботи передавача:

```
void loop() {
    int val = map(analogRead(pot),0,1024,0,255); //змінна, що зберігає стан резистора
    LoRa.beginPacket();  //початок пакету
    LoRa.print(val); //передавання пакету
    LoRa.endPacket(); //кінець пакету
    delay(50); //пауза 50мс між передачами пакетів
}
```

Код приймача:

Підключення бібліотек:

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

Опис виводу до якого приєднано світлодіод для тестування статусу приймання даних:

```
int LED = 3;
```

Опис змінних:

```
String inString = "";    // строка для зберігання отриманої інформації
int val = 0; //отримане числове значення, що передається про рівень опору резистору
```

Блок первинних налаштувань:

```
void setup() {
    Serial.begin(9600); //запуск серійного порту для передавання статусу передавача через USB
    pinMode(LED,OUTPUT); //вказання напрямку роботи виводу до якого підключено світло діод, що вказує статус прийому
    while (!Serial);
    Serial.println("LoRa Receiver");//повідомлення в порт, що приймач запущено
    if (!LoRa.begin(433E6)) { // або 915E6, швидкість у МГц для модуля
    Serial.println("Start LoRa error!");
    while (1);
    }
}
```

Основний блок перевірки роботи приймача:

```
void loop() {
    // намагаємося отримати пакет
    int packetSize = LoRa.parsePacket();//розмір отриманого пакету
    if (packetSize) { // читаємо пакет
        while (LoRa.available())
        {
            int inChar = LoRa.read();//читаємо числовий пакет
            inString += (char)inChar; //додаємо пакет до строки
            val = inString.toInt(); //переводимо значення в числовий вигляд
```

```
            }
            inString = "";
             LoRa.packetRssi();
        }
        Serial.println(val);  //виводимо в порт отримане значення
        analogWrite(LED, val); //вмикаємо світло діод з яскравістю, що відповідає отриманому значенню
}
```

Висновки

В кваліфікаційній роботі розроблено модуль керування доступу до ліфтів багатоквартирних житлових будинків, що дозволяє обмежувати доступ злісним неплатникам за обслуговання будинку.

Було проведено наступні роботи:

- проаналізовано стан систем автоматизації в сфері ЖКГ;

- проведено огляд існуючих систем керування доступом до ліфтів та наведено їх недоліки;

- розроблена структура системи керування доступом до ліфтів;

- обрано конструктивні елементи модулю контролю доступу та розроблено конструкцію з вказанням методики підключення для ліфтів різних модифікацій, що використовуються в Україні;

- розроблено структуру бази даних для зберігання інформації на сервері щодо користувачів системи для подальшої передачі на модуль керування доступом;

- розроблено протоколи обміну даними між модулем та вузлом територіального збору даних та між вузлом та сервером;

- розроблено програмне забезпечення мікроконтролера модуля.

Кожній людині видається особистий електронний ключ (картка, брелок або ідентифікаційний номер NFC мобільного телефону), за яким надається доступ до проїзду в ліфті у разі сплати за обслуговання будинку. У будь-який момент часу адміністратор може змінити права доступу, причому віддалено, тобто фізично апаратний ключ йому для цього не потрібен. Це зручно у випадках, коли йде ремонт - тоді будівельникам можна відразу видати спецкарточкі, які будуть працювати тільки певний період часу та/або лише у вантажному ліфту. А після ремонту ці картки будуть баніться - і проїзд по ним буде неможливий. Також зручно заблокувати картку в разі її втрати та видачі нової.

Контроль доступу в ліфт за електронними ключами при мережевий системі контролю доступу можна налаштувати так, що при піднесенні картки до зчитувача, система сама визначить, на який поверх відвезти користувача.

В першу чергу система призначена для обмеження доступу тим особам, яким не належить користуватися ліфтом. До них відносяться люди, які не проживають в будинку, а також, злісні неплатники. Важливою перевагою її використання є зниження актів вандалізму оскільки можливо визначити хто саме та коли їздив на ліфті.

Система являє собою комплекс технічних засобів і програмного забезпечення, що дозволяє експлуатувати обладнання з максимальною ефективністю завдяки використанню бездротового віддаленого адміністрування. Також система дозволяє збирати інформацію про осіб, які експлуатують ліфт. У багатьох будинках актуально обмеження маршрутів ліфтового обладнання, наприклад, зупинка на певних поверхах.

Перелік посилань

1. Baoan Li, Jianjun Yu. Research and application on the smart home based on component technologies and Internet of Things. // Procedia Engineering 2017. – 234 p.

2. Sajja A., Kharde D., Pandey C. A. Survey on efficient way to Live: Smart Home -It's an Internet of Things // ISAR -International Journal of Electronics and Communication Ethics 2016. – 26c.

3. Жежич П. І.Консолідовані інформаційні ресурси баз даних та знань : навчальний посібник / П. І. Жежич ; ред. В. В. Пасічник. -Львів : Вид-во Львівська політехніка, 2010. - 212 с.

4. Arduino відео-уроки. URL: https://arduinomaster.ru/ (дата звернення 26.04.2021)

5. Arduino tutorial. URL: https://www.arduino.cc/ (дата звернення 04.05.2020)

6. Все, що треба знати об Arduino.URL: http://arduino-diy.com (дата звернення 05.05.2021)

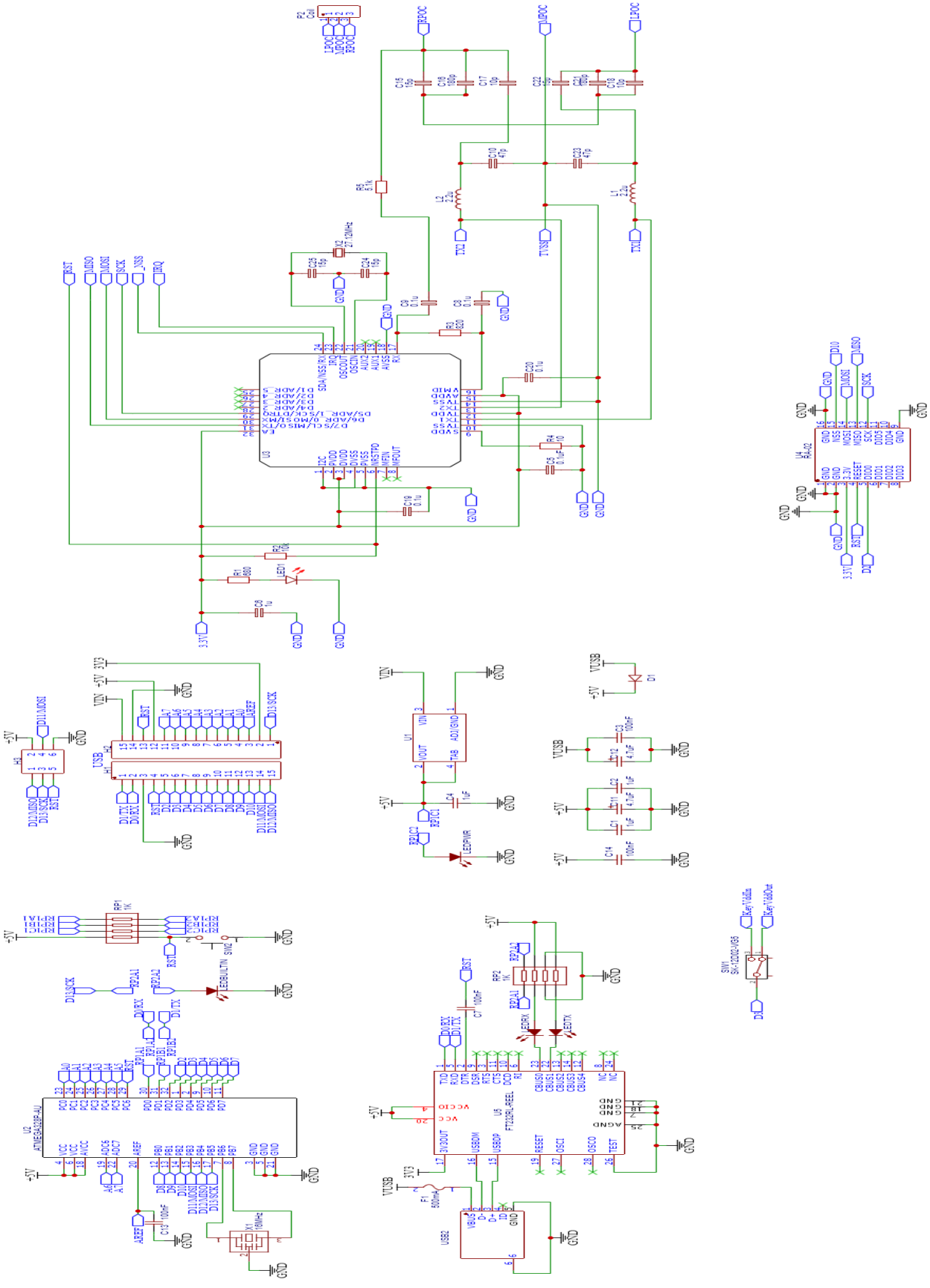
7. Переривання на мікроконтролері Arduino. URL: http://robotosha.ru/arduino/arduino-interrupts.html (дата звернення 05.05.2021)

8. Функції Arduino. URL: http://codius.ru/articles.html (дата звернення 05.05.2021)

9. Arduino Library for Proteus V2.0 URL: https://www.theengineeringprojects.com/2021/03/arduino-library-for-proteus-v2.html . (дата звернення 25.05.2021)

10. Онлайн середовище розробки EasyEDA URL: https://easyeda.com/ (дата звернення 26.05.2021)

Додаток А – Схема модулю



Додаток Б – програма керуючого мікроконтролера

```
#include <avr/eeprom.h> //Подключаем библиотеку по работе с ППЗУ
#include <SPI.h> //Подключаем библиотеку для работы по шине SPI
#include <MFRC522.h> //подключаем библиотеку для работу со считывателем RFID RC522 13.56MHz
#include <LoRa.h> //подключаем библиотеку по работе с модулем передачи LoRa (Lora By Sendeep Mistry)
//обозначаем выводы которые будут использоваться для управления модулями
#define RST1_PIN 7//RST контакт считывателя
#define SS1_PIN  8//SDA контакт считывателя
#define NSS 10 //NSS LoRa
#define DIO0 2 //DIO0 LoRa
#define Relay 3 //контакт управляющий замыканием реле клавиатуры
MFRC522 RFID_IN(SS1_PIN, RST1_PIN);  // Создаем объект считывателя входа
//объявляем переменные
unsigned long uidDec, uidDecTemp;  // для храниения номера метки в десятичном формате
byte LastuidDec[4]; //последняя считанная ИД карты по байтам
unsigned long StartSRead,StopSRead;//засекаем время начала и конца периода связи с териториальным узлом
byte MTm=150;//время (в секундах)связи с териториальным узлом
int packetSize=0; //размер полученного от узла пакета
byte EuipID[4]={0x00,0x00,0x01,0x01};//апаратный ИД данного экземпляра модуля

//блок первоначальных настроек
void setup() {
  //настройка направлений портов
  pinMode(Relay,OUTPUT);
  pinMode(NSS,OUTPUT);
  pinMode(DIO0,OUTPUT);
  //размыкаем на всякий случай реле
  digitalWrite(Relay,LOW);
  //запуск порта для тестирования в процесе отладки
  Serial.begin(9600);
  //инициализация считывателя
  SPI.begin();      // Инициализация шины SPI
  RFID_IN.PCD_Init();   // Инициализация считывателя
  //убеждаемся что считыватель подключен корректно
  if (GetRFIDVer()==0) {ShowRFID_INErr(); }
  //пытаемся запустить приемомередатчик
```

```
  if (!LoRa.begin(433E6)) {
    Serial.println("Start LoRa error!");
  while (1);
  }
  //выводим информацию, что модуль запущен
  Serial.println("Module Started");
  delay(500);
  StartSRead=millis();//начало периода засекания времени связи с узлом
 }

 //основной блок программы
 void loop() {
  StopSRead=millis();
  if ((StopSRead-StartSRead)>(MTm*1000)) { //если прошел необходимый период для связи
   StartSRead=millis(); //заново засекаем время для связи
   //отправляем запрос на проверку наличия обновления списка карт
   //02 ID ID ID ID 81 81 03
   byte LoraPkt[8]={0x02, EuipID[0], EuipID[1], EuipID[2], EuipID[3], 0x81, 0x81, 0x03};
   LoraSend(LoraPkt, 8); //отправляем запрос
   //ожидаем ответ
   byte readPkt[7]; //буфер приема
   byte n=0; //кол-во принятых байт
  waitPkt:
   packetSize = LoRa.parsePacket();
   if (packetSize) {
    while (LoRa.available())
    {
     readPkt[n]=LoRa.read();
     n++;
    }
   LoRa.packetRssi();
   }
   //убеждаемся что пакет не битый и для нас 02 ID ID ID ID FF/00 03
   if ((readPkt[0]==0x02)&&(readPkt[1]==EuipID[0])&&(readPkt[2]==EuipID[1])&&(readPkt[3]==EuipID[2])&&(readPkt[4]==EuipID[3])&&(readPkt[6]==0x03)) {
    if (readPkt[6]==0x00) { //если изменений не было - идем на начало программы
     ackChngs(); //отправлем подтверждение о полученой информации
     return;
    }
    if (readPkt[6]==0xFF) { //если изменения были - запрашиваем изменения
```

```
            ackChngs(); //отправлем подтверждение о полученой информации
            IdsNumberChng;//отправляем запрос о кол-ве ключей которые надо менять
             }
           }
         }
         if (!RFID_IN.PICC_IsNewCardPresent())
         { return; }     //если не поднесена к считывателю карта - переходим на начало программы
         if (RFID_IN.PICC_ReadCardSerial()) { //если поднесена карта - производим ее чтение и анализ
         // читаем карту с входа
          CardIDOut();  //выводим в порт ИД карты и принимаем решение о предоставлении доступа
         }
        }

        //отправляем запрос о кол-ве ключей которые надо менять
        void IdsNumberChng()
        {
         //02 ID ID ID ID 55 AA 03
          byte SendPkt[8]={0x02, EuipID[0], EuipID[1], EuipID[2], EuipID[3], 0x55, 0xAA, 0x03};
          LoraSend(SendPkt, 8); //отправляем запрос
        }

        //процедура отправки подтверждения о полученном статусе изменений
        void ackChngs()
        {
          byte SendPkt[8]={0x02, EuipID[0], EuipID[1], EuipID[2], EuipID[3], 0x0F, 0x81, 0x03};
          LoraSend(SendPkt, 8); //отправляем запрос
        }

        //процедура отравки пакета по приемопередатчику
        void LoraSend(byte* arrayPtr, byte arrSize) {
         LoRa.beginPacket();
         for (byte i=0; i<arrSize; i++) {
          LoRa.print(arrayPtr[i]); }
         LoRa.endPacket();
         delay(50);
        }

        //выводим в порт ИД карты и принимаем решение о предоставлении доступа
```

```
        void CardIDOut() {
          uidDec = 0;
          for (byte i = 0; i < RFID_IN.uid.size; i++)  {//читаем карту
              uidDecTemp = RFID_IN.uid.uidByte[i];
              uidDec = uidDec * 256 + uidDecTemp;
              LastuidDec[i]=uidDecTemp;//сохраняем последнюю считанную карту
            }
          Serial.print("RFID:"); //выводим ИД в порт
          Serial.println(uidDec);
          int acc=TestCard();
          if (acc==0) {//если карты даже нет в списке
            Serial.print("No Card In List!");
            digitalWrite(Relay,LOW);//размыкаем на всякий случай реле
            }
          if (acc==1) {//если карта есть в списке, но доступ ей запрещен
            Serial.print("Card Have not access!");
            digitalWrite(Relay,LOW);//размыкаем на всякий случай реле
            }
          if (acc==2) {//если карта есть в списке и доступ разрешен
            Serial.print("Card Have access! Keyboard ON");
            digitalWrite(Relay,HIGH);//замыкаем реле
            delay(15000); //15 секунд паузы чтоб пользователь успел нажать
кнопку
            }
          return;
        }

        //проверяем есть ли карта LastuidDec в памяти и имеет ли она доступ
        int TestCard() {
          //в цикле просмотриваем содержимое ППЗУ
          for (int i=0; i<100; i++) { //можем хранить до 100 карт
            //на каждую карту 5 байт 1й разрешение - AA, запрет - 55 или место
пустое - 00
            //еще 4 байта сам ИД карты
            unsigned int eeaddr=i*5;
            byte b1=eeprom_read_byte(eeaddr);
            if (b1==0x00) {return 0;}
            eeaddr++;
            byte b2=eeprom_read_byte(eeaddr);
            if (b2!=LastuidDec[0]) { continue;}
            eeaddr++;
            b2=eeprom_read_byte(eeaddr);
            if (b2!=LastuidDec[1]) { continue;}
            eeaddr++;
            b2=eeprom_read_byte(eeaddr);
```

```
    if (b2!=LastuidDec[2]) { continue;}
    eeaddr++;
    b2=eeprom_read_byte(eeaddr);
    if (b2!=LastuidDec[3]) { continue;}
    if (b1==0x55) {return 1;}
    if (b1==0xAA) {return 2;}
   }
   return 0; //если перебрали всю память и ничего не нашли - значит карты нет в списке
  }

  //выводим в порт ошибку подключения считывателя на входе
  void ShowRFID_INErr () {
   Serial.println("No RFID IN");
   delay(2000);
   return;
  }

  //проверить соединение со считывателем - получить его версию 1 или 2 либо
  //3 - версия не извесна; 0 - если соединение отсутствует
  byte GetRFIDVer() {
   byte v;
   // получить версию ПО MFRC522
   v = RFID_IN.PCD_ReadRegister(RFID_IN.VersionReg);
   // Если 0x00 или 0xFF - соединение отсутствует
   if ((v == 0x00) || (v == 0xFF)) {
    return 0;
   }
   if (v == 0x91)
    return 1;
   else if (v == 0x92)
    return 2;
   else
    return 3;
  }
```

Додаток В – Презентація