

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ»**

**Кафедра комп'ютерної інженерії**

ДО ЗАХИСТУ ДОПУЩЕНА

Зав. кафедрою \_\_\_\_\_

д.т.н., проф. Переверзєв А.В.

**БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА**

**КЛІЄНТ-СЕРВЕРНА СИСТЕМА КОНТРОЛЮ УПРАВЛІННЯ ДОСТУПОМ  
СКЛАДСЬКИХ ПРИМІЩЕНЬ**

Виконав

ст. гр. КІ-128

\_\_\_\_\_

А.К. Рассохін

Керівник

доцент

\_\_\_\_\_

О.А. Жеребцов

Запоріжжя

2022

**ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ»**

**Кафедра комп'ютерної інженерії**

ЗАТВЕРДЖУЮ

Зав. кафедрою

д.т.н., професор

А.В. Переверзев

---

17.01.2022 р.

**З А В Д А Н Н Я  
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ**

Студенту гр. \_\_\_\_\_ *KI-128*, спеціальності *123 «Комп'ютерна інженерія»*

\_\_\_\_\_ *Рассохіну Арсенію Костянтиновичу*

1. Тема: Клієнт-серверна система контролю управління доступом складських приміщень

затверджена наказом по інституту 06.1-50 від 15 січня 2022 року.

2. Термін здачі студентом закінченої роботи: 18 червня 2022 року.

3. Перелік питань, що підлягають розробці:

1. Дослідити предметну область систем контролю управління доступом.
2. Здійснити огляд існуючих систем контролю та проаналізувати принцип їх структурної побудови.
3. Розглянути можливість запровадження клієнт-серверної системи контролю управління доступом складських приміщень.
4. Здійснити вибір програмно-апаратного комплексу для реалізації проекту.
5. Розробити апаратне рішення електронного вузла виконавчого пристрою, спираючись на обрану базу компонентів.
6. Розробити програмний комплекс керування виконавчим пристроєм.
7. Розробити програмний алгоритм прийому, передачі та керування розгорнутої серверної частини.
8. Розробити програмно-апаратне рішення пристрою внесення та зміни ідентифікаційних даних до бази даних.
9. Розробити графічну панель керування системою контролю управління доступом.
10. Провести контрольне налаштування та випробування системи.
11. Оформити результати роботи у вигляді звіту.

Дата видачі завдання: 17 січня 2022 р.

Керівник бакалаврської роботи

\_\_\_\_\_

О.А. Жеребцов

Завдання отримав до виконання

\_\_\_\_\_

А.К. Рассохін

## РЕФЕРАТ

Бакалаврська дипломна робота містить: 123 сторінки, 90 рисунків, 14 таблиць, п'ять додатків, 43 першоджерела.

Мета роботи – створення клієнт-серверної системи контролю управління доступом складських приміщень, на базі взаємодії периферійних пристроїв з базою даних за допомогою сокет з'єднань.

Предмет дослідження – дослідження інформаційних процесів взаємодії у системі з клієнт-серверною архітектурою.

Об'єкт дослідження – автоматизована системи контролю та управління доступом.

У першому розділі проведено дослідження існуючих систем контролю доступу. Розглянуто актуальні методи побудови систем контролю та управління доступом у побутових та промислових випадках застосування. Обґрунтовано рішення розробити і запровадити клієнт-серверну систему контролю та управління доступом, основою якої є, керуючий периферійними пристроями, одно кристальний комп'ютер та сервер, що розгорнутий за допомогою робочої станції, які об'єднані в локальну мережу.

У другому розділі описано та обґрунтовано вибір програмних та апаратних інструментів розробки для реалізації клієнт-серверного програмно-апаратного рішення.

У третьому розділі розглянутий процес розробки і проектування апаратної та програмної частини проекту системи контролю доступу, а також, розглянуто процес тестування та налагодження системи.

Результати роботи були оприлюднені на тринадцятій всеукраїнській конференції «Актуальні проблеми математики та інформатики».

СКУД, СЕРВЕР, КЛІЄНТ, ТСР-З'ЄДНАННЯ, СОКЕТ, ОБМІН ДАНИМИ,  
МІКРОКОНТРОЛЕР, STM, MySQL,.NET

## ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	<b>Error! Bookmark not defined.</b>
РОЗДІЛ 1 ОГЛЯД СИСТЕМ КОНТРОЛЮ ТА УПРАВЛІННЯМ ДОСТУПОМ .....	11
1.1 Класифікація СКУД .....	12
1.1.1 Класифікація СКУД за кількістю пунктів пропуску .....	14
1.1.2 Класифікація СКУД за різновидом пристрою зчитування.....	14
1.1.3 Класифікація СКУД за різновидом виконавчого пристрою.....	15
1.1.4 Класифікація СКУД за рівнем фізичного захисту .....	18
1.1.5 Класифікація СКУД за методом мережевої комутації.....	19
1.2 Огляд технології радіочастотної ідентифікації .....	20
1.3 Огляд популярної СКУД .....	24
1.4 Створення принципової моделі проекту .....	27
1.5 Висновки за розділом .....	28
РОЗДІЛ 2 .....	30
ВИБІР ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ПРОЕКТУ .....	30
2.1 Вибір та обґрунтування базових вузлів для створення апаратного комплексу проекту .....	30
2.1.1 Контролер виконавчого пристрою .....	30
2.1.2 Контролер радіочастотної ідентифікації .....	36
2.1.3 Чіп зберігання ідентифікаційних даних .....	39
2.1.4 Контролер з'єднання мережі .....	43
2.1.5 Контролер пристрою внесення даних.....	47
2.2 Вибір та обґрунтування базових рішень для створення програмного комплексу проекту .....	50
2.2.1 Програмне забезпечення для програмування блоку виконавчого пристрою.....	51
2.2.2 Програмне забезпечення для програмування блоку серверної частини.....	62
2.3 Висновки за розділом .....	68
РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РОЗРОБКА ТА ВИПРОБУВАННЯ .	69

3.1 Проектування СКУД.....	69
3.2 Проектування виконавчого пристрою .....	69
3.3 Проектування пристрою внесення даних .....	78
3.4 Програмування комплексу виконавчого пристрою .....	80
3.5 Програмування обробника бази даних .....	91
3.6 Програмування пристрою внесення даних .....	101
3.7 Програмування додатку роботи з даними співробітників.....	103
3.8 Тестування та налагодження СКУД.....	110
3.9 Висновки за розділом .....	116
ВИСНОВКИ.....	118
ПЕРЕЛІК ПОСИЛАНЬ.....	119
Додаток А. Принципові електричні схеми .....	<b>Error! Bookmark not defined.</b>
Специфікація бази електронних компонентів.....	<b>Error! Bookmark not defined.</b>
Додаток Б. Моделі друкованих плат .....	<b>Error! Bookmark not defined.</b>
Додаток В. Вихідний код програми мікроконтролера .....	<b>Error! Bookmark not defined.</b>
Додаток Г. Вихідний код серверу обробки даних .....	<b>Error! Bookmark not defined.</b>
Додаток Д. Вихідний код клієнтської частини ..	<b>Error! Bookmark not defined.</b>
Файл MainForm.cs .....	<b>Error! Bookmark not defined.</b>
Файл FormDesigner.cs .....	<b>Error! Bookmark not defined.</b>
Файл DataBank.cs .....	<b>Error! Bookmark not defined.</b>
Файл FormContact.cs .....	<b>Error! Bookmark not defined.</b>
Файл FormContactDesigner.cs .....	<b>Error! Bookmark not defined.</b>

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

Слово/Словосполучення	Скорочення
А	
АЛП	Алгоритмічно-логічний пристрій
АЦП	Аналогово-цифровий перетворювач
Д	
ДП	Друкована плата
О	
ОЗП	Оперативно-запом'ятовуючий пристрій
ОС	Операційна система
П	
ПЗП	Постійний-Запам'ятовуючий Пристрій
ПІБ	Прізвище Ім'я По-батькові
С	
СКУД	Система Контролю та Управління Доступом
СУБД	Система Управління Базами Даних
Ц	
ЦАП	Цифро-Аналоговий Перетворювач
Е	
ЕОП	Електронно-Обчислювальний Пристрій
А	
ARP	Address Resolution Protocol
С	

CAN	Control Area Network
CMSIS	Common Microcontroller Software Interface Standard
CRC	Cyclic Redundancy Check
D	
DMA	Direct Memory Access
E	
EEPROM	Electrically Erasable Programmable Read-Only Memory
F	
FIFO	First In First Out
G	
GPIO	General Purpose Input-Output
GUI	Graphical User Interface
H	
HAL	Hardware Abstraction Layer
I	
IC	Integrated Circuit
ICMP	Internet Control Message Protocol
IDE	Integrated Development Environment
IGMP	Internet Group Management Protocol
IOT	Internet Of Things
I/O	Input-Output
IP	Internet Protocol
L	
LINQ	Language Integrated Query
LoRa	Long Range Protocol
M	
MCU	Microcontroller Unit
MISO	Master In Slave Out
MOSFET	Metal-Oxid-Semiconductor Field-Effect-Transistor



MOSI	Master Out Slave In
MPU	Microprocessor Unit
N	
NFC	Near Field Communication
NXP	Nexperia
O	
OSI	The Open Systems Interconnection
P	
PHY	Physical Layer
POS	Point Of Sale
PWM	Phase Width Modulation
R	
RAM	Random Access Memory
RFID	Radio-Frequency Identification
RNG	Random Ranking Generator
RX	Receive
S	
SMD	Surface-Mounted Device
SPI	Serial-Peripheral Protocol
SRAM	Static Random Access Memory
SSL	Secure Sockets Layer
T	
TCP	Transmission Control Protocol
TX	Transmit
U	
UDP	User Datagram Protocol
UID	Unique Identifier
USART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
W	

WI-FI	Wireless Fidelity
WOL	Wake On LAN

## ВСТУП

Розвиток наукової сфери проектування та будування комп'ютерних систем, знайшов свій шлях у потребі розробки методів взаємодії різноманітних систем різного рівня інженерної складності. Так з'явилися електронно-обчислювальні прилади, що побудовані на одному кристалі та, згідно завдання, які займають мало місця й потребують менше ресурсів та складності під час виготовлення- мікроконтролери. Вони зайняли своє почесне місце в сучасних інженерних розробках та в популярній на сьогоднішній день течії- вбудованих системах (англ. Embedded System).

Вбудована система- спеціалізована мікропроцесорна система управління, контролю та моніторингу, концепція розробки якої полягає в тому, що така система буде працювати, будучи вбудованою безпосередньо у пристрій, яким вона керує.

Вбудовані системи використовуються в якості як відокремлених електронних вузлів або систем, так і в умовах ланцюгового з'єднання, з використанням різних видів та методів зв'язку.

Основними напрямками використання структур вбудованого типу є: засоби автоматичного регулювання та управління технологічними процесами, наприклад авіоніка, контроль доступу, верстати з ЧПУ (Числовим Програмним Керуванням), банкомати, платіжні термінали, телекомунікаційне обладнання та величезна кількість інших реалізацій будь-якого типу та належності.

Завдяки великій кількості переважаючих властивостей мікроконтролерної невеликогабаритної техніки, вона зайняла суттєву нішу в програмі розробок електронних інженерів в усьому світі.

## РОЗДІЛ 1

### ОГЛЯД СИСТЕМ КОНТРОЛЮ ТА УПРАВЛІННЯМ ДОСТУПОМ

За визначенням вікіпедії СКУД це комплекс технічних та програмних засобів безпеки, що здійснює регулювання входу / виходу та переміщень людей чи транспортних об'єктів на територіях, які знаходяться під охороною, для адміністративного моніторингу та попереджень несанкціонованого проникнення [1].

На даний час вже існує безліч реалізацій систем контролю, що давно вийшли за межі їх базової функції- надання доступом до чогось або до когось за наявністю прав.

Світовий ринок подібних систем здатен задовільнити будь-якого прискіпливого замовника, та навіть більше, демонструючи різноманітні цінові та якісні пропозиції, починаючи від бюджетних пристроїв локального використання, де не потрібен високий фізичний та програмний рівень захисту до доволі недешевих клієнт-серверних систем, що спроектовані безперервно охороняти фінансові заклади або об'єкти стратегічного значення.

Ідентифікатором особи можуть слугувати як біометричні дані, такі як: райдужка ока, відбиток пальців, форма лиця, форма долоні та голос, так і більш загальноприйняті, ті, які гарно себе зарекомендували- дані електронного носія, який комунікує за допомогою радіо-сигналу або його емуляції за допомогою спеціального або загально-прийнятого алгоритмів. Інколи, до вище запропонованих методів додаються інтернет додатки, де можна керувати доступом об'єкту, який треба охороняти.

Найважливішим та базовим завданням на початку створення прототипу системи контролю доступом є вибір програмної та апаратної структури, що заздалегідь повинна фігурувати у технічному завданні проекту.

Насамперед, треба чітко оглянути існуючі аналоги подібних систем та віднести систему, що створюється, до окремої категорії за її характеристиками

та технічними можливостями. Для цього має бути визначені потреби від СКУД, а саме: місце її використання, кількість користувачів, фізична та програмна стійкість системи за умов її використання.

Створення коректного плану полегшить роботу під час проектування та найголовніше- покращить результат вже готового прототипу, виключаючи велику кількість різноманітних проблем під час проектування за наявності якісного аналізу потреб на початку.

## 1.1 Класифікація СКУД

Системи контролю управління доступом поділяються на два базових різновиди за системною топологією, від яких надалі відбувається розширення властивостей: автономні та мережеві.

1) Автономні системи - це системи, що мають здатність контролювати та вести запис безпосередньо за допомогою виконуючого пристрою, що територіально знаходиться біля зчитувача (рис. 1.1).

До переваг системи відноситься її помітна простота та невелика вартість, те, що вона не потребує апаратних рішень, що можуть вплинути на кінцеву складність, таких як: серверна частина, система комунікацій, програмних реалізацій серверної частини. Вона підходить до використання на невеликих об'єктах та там, де відповідальним та контролюючим охороною об'єктом не треба постійно звертатись до виконавчого пристрою віддалено, задля корегування його роботи, слідкуванням за подіями та видаленням або додаванням нових ідентифікаційних орієнтирів.

Негативні характеристики подібних систем є, але вони доволі суб'єктивні, адже відображають окремий клас приладів та не зовсім коректні до порівняння із мережевими клієнт-серверними системами. Тим не менш: простота часто впливає на функціональні особливості, отже автономні системи не завжди можуть задовільнити потребам користувача. Також,

подібні пристрої не завжди мають гарну якість виконання як апаратної, так і програмної реалізації, в більшості не мають криптографічного захисту.

## **КЛІЄНТ** (Виконавчо/оброблюючий пристрій)

- 1) Зчитування ID
- 2) Обробка
- 3) Зберігання
- 4) Внесення/видалення ID
- 5) Надання фізичного доступу

Рисунок 1.1 – Позначення вузла клієнт

2) Мережеві системи- це системи з більш розширеним та наповненим різноманітними корисними можливостями функціоналом. Такі системи створені аби мати можливість вести повний контроль та керування їм за допомогою комп'ютеру з різних, нефіксованих місць, таких як: відділ кадрів, офіс та навіть маніпулювати з телефону.

Мережеві системи використовуються на великих об'єктах та підприємствах, де основна вимога- вести керування з єдиного пункту управління, та зберігати всі дані саме на ньому. Подібна система за своїми позитивними рисами виграє у систем автономного типу, але користувачу треба заздалегідь мати на увазі більшу вартість такого рішення.

## **КЛІЄНТ** $\longleftrightarrow$ **СЕРВЕР** (Виконавчий пристрій) (Оброблюючий пристрій)

- |                              |                             |
|------------------------------|-----------------------------|
| 1) Зчитування ID             | 1) Відправка/прийом (даних) |
| 2) Відправка/прийом (даних)  | 2) Обробка                  |
| 3) Надання фізичного доступу | 3) Зберігання               |
|                              | 4) Внесення/видалення ID    |

Рисунок 1.2 – Позначення вузлів клієнт та сервер

Окрім цього базового поділення також існує декілька класифікацій СКУД, які необхідно розглянути нижче.

### 1.1.1 Класифікація СКУД за кількістю пунктів пропуску

За кількістю пунктів пропуску системи контролю керування доступом поділяються на три різновиди. Кожен з них має переваги та недоліки один перед одним, класифікація залежить від чіткої потреби пункту, щодо потрібної кількості пунктів керування фізичного доступом на ньому. Поділяються на:

- 1) До 16 пунктів пропуску - автономні і мережеві системи контролю доступом. Характеризується невеликою та, у більшості, нерозгалуженою системою мережевих комунікацій. Об'єкти використання: приватні будинки, невеликі навчальні заклади, невеликі підприємства, та інші. ;
- 2) Від 16 до 64 пунктів пропуску- мережеві системи контролю доступом. Характеризуються більш великою за перший рівень мережею комунікаційних з'єднань. Об'єкти використання: невеликі або великі навчальні заклади, невеликі або середні підприємства, та інші;
- 3) Від 64 пунктів пропуску- мережеві системи контролю доступом. Нічим не відрізняється від другого рівня, окрім більш великої мережі комунікаційних з'єднань. Об'єкти використання: великі навчальні заклади, великі підприємства, та інші об'ємні структурні підрозділи.

### 1.1.2 Класифікація СКУД за різновидом пристрою зчитування

За різновидом пристрою зчитування системи контролю керування доступом поділяються на три основних різновиди:

- 1) Біометрична- система, що виконує ідентифікацію людини шляхом зчитування його індивідуальних фізичних рис. Існують такі методи індивідуальної ідентифікації людини: 3-х вимірна фотографія обличчя, фотографія та ідентифікація райдужки ока, зчитування особливостей відбитків пальців та інші.

Переваги: відсутність потреби мати при собі пристрій-носії ідентифікаційної інформації, майже унеможлиблює процес підробки ідентифікатора.

Недоліки: менш швидкий процес ідентифікації, більша кінцева вартість системи, через використання складних оптичних або резистивних приладів зчитування.

2) Цифрова (RFID)- система, що ідентифікує людину шляхом порівняння цифрової інформації, яка знаходиться на електронному носії з тією інформацією, що знаходиться в базі зберігання даних користувачів електронних носіїв. Передача даних з міток створюється методом передачі радіосигналів (або емуляції радіосигналів) від мітки до зчитувача.

Переваги: невелика вартість носіїв інформації та системи зчитування загалом, особливо в порівнянні з біометричною, гарні криптографічні властивості зберігання інформації, велика швидкість зчитування ідентифікатора.

Недоліки: потреба мати картку-носії завжди при собі, теоретична можливість несанкціонованого зчитування інформації з носія.

3) Кодова- найбільш проста та застаріла система ідентифікації, що в сьогоденні майже не використовується окремо від двох перших. Ідентифікатором є код-пароль, що користувач має ввести на пристрої вводу.

Переваги: мала вартість системи, відсутність потреби мати ідентифікаційний носій.

Недоліки: потреба для користувача пам'ятати унікальний код доступом, ускладнення процесу ідентифікації окремих осіб та ведення табелю обліку робочого часу, погана стійкість до зламування.

### 1.1.3 Класифікація СКУД за різновидом виконавчого пристрою

Виконавчий пристрій системи контролю- це фізичний механізм, що призначений обмежувати доступ людини або предмету на територію, що



пристрій охороняє. Залежно від об'єкту та від рівня його захищеності, де механізм буде працювати систему контролю керування доступом поділяються на 3 види:

1) Турнікети - найбільш популярний тип захисного механізму пунктів пропуску підприємств, метрополітену та інших об'єктів, де відбувається великий потік пересування людей, що контролюються. Вони найбільш продуктивно працюють в тих місцях, де потрібна швидкість перевірки та пересування.

Турнікетні системи зазвичай поділяються на 3 основні типи:

- Полу ростові триподні турнікети- турнікетні системи, де обмежуючим пристроєм є трипод (3 ригелі, що спрямовані один відносно іншого під кутом 120 градусів). кінематика приладу сконструйована таким чином, щоб тільки одна людина могла пройти скрізь ного за один такт пропускання. Частіше за все, подібні турнікетні системи зустрічаються на підприємствах, в офісних центрах та навчальних закладах.



Рисунок 1.3 – Зображення полу ростових турнікетів

- Повно ростові триподні турнікети- турнікетні системи, що за методом дії схожі із полу ростовим турнікетом, але мають більш гарні показники безпеки, адже виключають можливість проникнення людини зверху, якщо та намагатиметься це зробити.



Рисунок 1.4 – Зображення повно ростових турнікетів

- Стулчасті системи- зазвичай зустрічаються у системах пропуску метрополітену, мають найгарніший показник пропускної здатності, адже механізм дуже простий до відкривання.



Рисунок 1.5 – Зображення стулчастих систем

2) Ригельні електромеханічні замки- електромеханічна система, що мало чим відрізняється від звичайного ригельного замка в побутових квартирах, крім того, що обладнані електронною системою контролю та без ключовим доступом. Мають гарні показники захисту, адже зазвичай вмонтовані в металеві двері або ставні. Поділяються на 2 основні типи:

- Механічні - приводом прямої дії є механічна система в невеликим електродвигуном, що змушує ригель рухатись;
- Соленоїдні- приводом прямої дії є електричний магніт, що змушує ригель рухатись.



Рисунок 1.6 – Зображення замку ригельного типу

1) Електромагнітні замки- це система, що складається з електричного магніту, що керується електронною частиною системи контролю керування доступом. Гарно спроектовані системи захисту, що використовують електромагнітний метод захисту дуже добре протидіють злому, адже знаходяться з іншої від зловмисника сторони двері або ставнів. Як і ригельні електромеханічні замки, електромагнітні неефективні у випадках, коли на об'єкті великий потік людей на пункті пропуску.



Рисунок 1.7 – Зображення замку електромагнітного типу

#### 1.1.4 Класифікація СКУД за рівнем фізичного захисту

За рівнем фізичного захисту системи контролю керування доступом поділяються на 2 основні рівня:

1) Перший рівень захисту за побудовою конструкції теоретично дозволяє несанкційно проникнути людині, що не має права доступом, тому зазвичай на практиці супроводжується контролерами або охоронцями на контрольних пунктах пропуску.

До цього рівня належать полу ростові турнікетні системи та стулчасті системи.

2) Другий рівень механічно запобігає несанаційному проникненню людини, адже за побудовою конструкції не залишає незахищеного місця. Охоронні системи цього рівня не потребують обов'язкової присутності охоронця на пункті пропуску.

До цього рівня належать: повно ростові турнікетні системи, ригельні електромеханічні замки та електромагнітні замки.

#### 1.1.5 Класифікація СКУД за методом мережевої комутації

Спосіб комутації клієнт-серверних систем контролю управління доступом визначає стійкість системи до спроб програмного злому та до електромагнітної або електричної стійкості. Метод підключення та його реалізація визначає експлуатаційні параметри системи загалом. На сьогоднішній день існує 2 методи:

1) Безпроводний- метод, що полягає у повній відсутності електричних кабелів зв'язку між виконавчим пристроєм на пункті пропуску та клієнтом, що бажає здійснити зв'язок із першим.

Загалом, метод використовує технології безпроводного з'єднання: АЕЕЕ 802.11 WI-FI, новішу LoRa та інші.

Основні переваги: повна відсутність кабелів дозволяє уникнути складної процедури протягнення кабельної мережі від одного до іншого, тим самим посилюючи безпеку фізичного впливу на процес передачі.

Основні недоліки: безпроводні системи, а, точніше, надійність підключення безпроводних систем у деяких випадках поступаються мережевому кабелю надійністю каналу зв'язку. Спеціальне створення електромагнітних перешкод може сприяти неповній передачі даних а інколи і повному зриву.

2) Кабельний- метод, що полягає в підключенні виконавчого пристрою на пункті пропуску до клієнта, або вузла розподілення а потім клієнта, методом протягування кабелю.

Загалом, професійні системи контролю використовують стандарти фізичного рівня: RS-232, RS-485 та Ethernet.

Основні переваги: надійний зв'язок та забезпечення повної передачі на прийому даних, стійкість до електромагнітних утручань.

Основні недоліки: потреба прокладання довгих кабелів, що треба ховати задля її безпеки від фізичного впливу, у деяких випадках - потреба у вузла комутації різних каналів.

## 1.2 Огляд технології радіочастотної ідентифікації

Більш щільно треба оглянути технологію радіочастотної ідентифікації, яким чином вона працює, де та як саме вона використовується, які її переваги та недоліки та чому саме її було обрано використовувати в проекті системи контролю та управління доступом складських приміщень.

RFID (Radio frequency identification) або радіо-частотна ідентифікація супроводжує історію ще з 1937 року, коли воєнні авіаційні системи навігації вперше використали спосіб розпізнавання «свій-чужий». Вона активно використовувалася під час другої світової війни. Перша демонстрація сучасних RFID-мікросхем як активного так і пасивного типу датується 1973 роком. Портативна система працювала на частоті 915МГц та використовувала 12-розрядні мітки. Аж у 1999 році була створена «глобальна система стандартів» для мікросхем RFID.

У сьогоденні системи радіочастотної ідентифікації використовуються в усіх сферах життя, починаючи від автоматизації виробництва, де за допомогою технології спрощуються та прискорюються процеси всіх видів виробництва, закінчуючи оплатою продуктів в магазині за допомогою NFC на POS-терміналі та, авжеж, системами контролю та управління доступом (СКУД).

Технологій RFID настільки багато, що, нажаль, вмістити їх в одну тему неможливо, тож потрібно вказати хоча б основні її види, що мають відношення

до ідентифікації людини для подальшого опрацювання процесу за допомогою отриманої інформації.

Сучасні системи контролю та управління доступом будь-яких об'єктів використовують дві основні ідентифікаційні технології: EM Marine та Mifare (рисунок 1.8).



Рисунок 1.8 – Логотипи технологій

1) EM Marine- найбільш популярна технологія зв'язку та зберігання інформації за допомогою «смарт-карт». Формат був розроблений швейцарською компанією EM Microelectronics. Технологія була розроблена перш за Mifare, та стала популярною у використанні з домофонними системами контролю побутових будинків в Україні.

Можна зустріти дві основних мікросхеми, на яких базуються прилади зберігання, вони працюють за основними стандартами частот 125КГц: ТК4100 та EM4100.

Чіп ТК4100 не є оригінальним продуктом компанії та використовується великою кількістю сторонніх компаній за однією технологією, встановлений в більшість карт, що існують.

Чіп EM4100 є оригінальним продуктом компанії EM Microelectronics та не є дуже розповсюдженим через значну вартість.

Переваги: невелика ціна, тому велика популярність та розповсюдженість. Через це не є проблемою знайти різноманітні прилади систем контролю, які будуть сумісні в роботі с мікросхемами ТК4100 та EM4100.

Недоліки: відсутній будь-який захист від перезапису та зчитування. Для цього не знадобиться складних на дорогих приладів для створення операції, неунікальний номер ідентифікації, котрий інколи написаний на передній частині «тіла» карти або електронного ключа, відсутність технології анти колізії (механізму запобігання зіткнення двох або більше ідентифікаторів), що при наявності їх поблизу одна одної під час зчитування буде викликати помилку.

2) Mifare- торгівельна марка розповсюдженої в Україні технології безконтактного зчитування за допомогою «смарт-карт» або електронних ключів що належить нідерландській компанії-виробнику напівпровідникової електроніки Nexperia (NXP) Semiconductors. Окрім NXP, ідентифікатори за ліцензією випускає компанія Infineon, тільки ті карти, на яких є позначки цих виробників можна вважати оригінальними, тільки вони слідкують за унікальністю номерів ідентифікації. Однак, як і в попередньому варіанті з EM Marine, Mifare виробляється й сторонніми компаніями.

Технологія отримала велику популярність у використанні із різноманітними СКУД, від малих підприємств до систем контролю стратегічно-важливих об'єктів, та завдяки своїм перевагам працює і надалі. Мікросхеми працюють за основними стандартами частот 13.56МГц та мають декілька серійних різновидів, від яких залежить рівень криптографічного захисту: Mifare Ultralight, Mifare Classic, Mifare Plus, Mifare DESfire, Mifare SmartMX.

Переваги: присутність захисту різних рівнів від перезапису та зчитування, велика внутрішня пам'ять пристрою, завдяки якій можна вносити записи різноманітної інформації та ключів доступу, механізм анти колізії, що унеможливорює «зіткнення» двох або більшої кількості ідентифікаторів у створі зчитувача, унікальні номери кожної окремої мікросхеми (UID), більша швидкість обміну інформацією між транспондером та блоком зчитувача.

Недоліки: більша вартість «смарт-карт» або електронних ключів в порівнянні з EM Marine, складніший процес запису/перезапису та зчитування, що інколи важливо.

Висновок: спираючись на публічний аналіз використання та особистий аналіз документації мікросхем радіочастотної ідентифікації компаній EM Marine та Mifare, що використовуються в системах контролю та управління доступом, були порівняні властивості двох технологій.

Як сучасну модель безконтактного зв'язку для використання в СКУД складських приміщень було обрано технологію Mifare від NXP, через вищий рівень функціональних особливостей та якіснішу реалізацію захисту.

Нижче приведено таблицю 1.1 порівняння основних характеристик транспондерів EM Marine та Mifare (Classic) версії.

Таблиця 1.1 – Порівняльна характеристика EM Marine та Mifare

	EM Marine	Mifare (Classic)
Довжина серійного номеру	3 байта	4/7 байт
Робоча частота	125 КГц	13,56 МГц
Присутність пам'яті	-	1/4 КБ
Присутність криптографічного захисту	-	+ Crypto 1
Режим роботи	Тільки зчитування	Зчитування/запис
Дальність зчитування	10 см	До 6 см
Можливість програмування	-	+ Біля 100 000 циклів
Захист від копіювання	-	+
Застосування	Прості СКУД	Складні СКУД



Окрім порівняння систем важливо зазначити їх суміжне різноманіття в апаратній реалізації. «Смарт-карти» бувають активними та пасивними.

Пасивні радіочастотні прилади- це прилади, що використовують зовнішню енергію, що надається від основного радіопередавача для забезпечення живлення для себе для обробки інформації. Загалом, дальність зчитування транспондерів від таких передавачів не більше 10ти сантиметрів, тому що такі карти не здатні передавати сигнал, він емулюється тактовими паузами, що створює мікросхема транспондера, коли живиться від передавача. Останній ці паузи зчитує та ініціалізує.

Активні радіочастотні прилади - це прилади, у яких за схемотехнічною побудовою є свій елемент живлення, що знаходиться поблизу чипа «смарт-карти». Подібні системи мають змогу виступати передавачем, та передавати інформаційний кодовий сигнал більш ніж на 10 метрів.

### 1.3 Огляд популярної СКУД

Однією з найбільш популярних компаній, що спеціалізується на проектуванні та виробництві систем контролю та управління доступом є «Bosch». Німецький гігант розробляє велику кількість різновидів СКУД, різного рівня захисту, екстраполяції та розгалуження, починаючи від локальних мережевих систем контролю, закінчуючи потужними контролерами для великих підприємств та установ.

Огляд системи від німецького виробника зроблено саме через скупчення гарних характеристик, показників та інженерних рішень в цілому. Проаналізувавши систему можна отримати деякі «підказки» та поради до проектування прототипу свого приладу.

«Access Easy Controller 2.1 (AEC2.1)» від «Bosch Security System»- це нове покоління контролера доступу, який поєднує в собі можливості вбудованого веб-сервера та системи відеоспостереження.

В цілому, хоч система найбільш схожа за своїм класом із системою, що проектується, але вона є головним контролером, що під'єднує до себе та комутує виконавчі пристрої різних пунктів надання доступу. Це відрізняє її за принципом роботи, адже система, що проектується, має бути відокремленим виконавчим пристроєм та не має бути залежною від центрального контролера, окрім серверу, що розгортається на ІВМ-подібній машині.

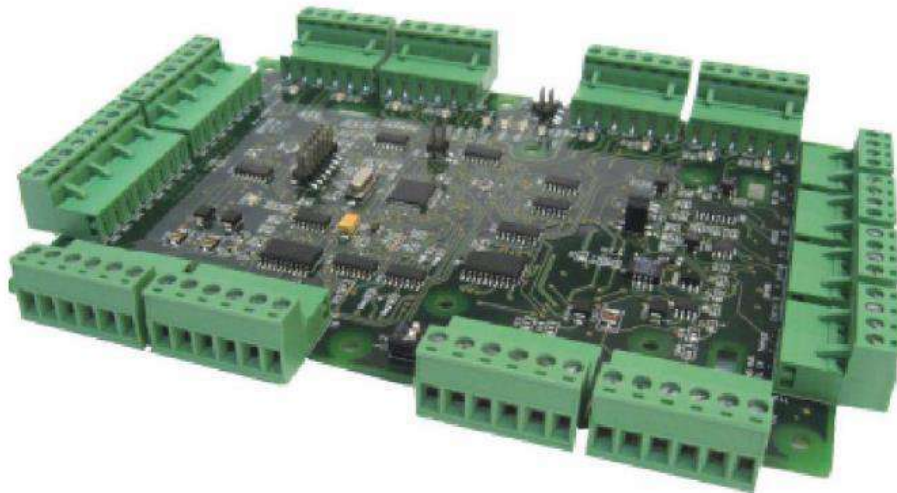


Рисунок 1.9 – Зображення контролера СКУД «Access Easy Controller 2.1 (AEC2.1)»

Особливості СКУД для адміністратора та співробітника:

- Кількість облікових записів адміністратора до 25 одиниць;
- 128-розрядний SSL алгоритм шифрування веб-системи керування;
- Захист облікових записів співробітників. Усі ідентифікатори користувачів і паролі зашифровані, забезпечуючи доступ лише для авторизованих користувачів для моніторингу та керування параметрів системи, записи транзакцій та діяльності;
- Має в наявності вбудований веб-сервер, за допомогою якого можна вести корегування та налаштування системи, використовуючи будь-який веб-браузер;
- Має вихід Ethernet, що дозволяє під'єднувати систему до будь-якої ІВМ-подібної машини.

- Види методів ідентифікації співробітників в системі:
  1. Радіочастотна ідентифікація з використанням 4-х або 7-ми байтного номеру;
  2. Ідентифікація за допомогою кодового паролю;
  3. Об'єднаний вид ідентифікації, що дозволяє використовувати два вище вказаних методи одночасно.

Основні характеристики СКУД «Access Easy Controller 2.1 (AEC2.1)»:

- Центральний процесор: 32-розрядний мікропроцесор із швидкістю тактування ядра 500МГц;
- Внутрішня пам'ять: 512МБ (RAM) та 512МБ (Flash);
- Напруга живлення: 12 В постійного току;
- Порти вводу-виводу: RJ45-Ethernet та RS-232;
- Кількість зчитувачів можливих до підключення: до 4 (2 двонаправлених виконавчих пристроїв);
- Кількість систем, що у випадку масштабування можна підключити у єдину систему контролю: до 32 одиниць контролерів;
- Кількість радіо ідентифікаційних транспондерів можливих до додавання: до 20 480 одиниць;
- Кількість можливих груп та ієрархій співробітників, що поділяють транспондери за привілеями доступу: до 254 груп;
- Можливість логування даних подій: присутня;
- Кількість пунктів можливого відеоспостереження: до 64 пунктів;
- Можливість SMS або e-Mail сповіщення, у нестандартних випадках;
- Можливість перегляду пунктів, на яких є відеоспостереження у реальному часі: присутня;
- Підтримка різних мов у веб-додатку: присутня;
- Можливість підключення вихідних та вхідних органів: до 64 одиниць контролюючих датчиків або сигналізації типу гучномовця.



Рисунок 1.10 – Зображення панелі керування СКУД «Access Easy Controller 2.1 (AEC2.1)»

Окремо можна виділити можливість додавання фотографій осіб співробітників під час створення їх індивідуальних сторінок. Це набагато полегшить роботу для контролюючої особи.

#### 1.4 Створення принципової моделі проекту

Здійснивши класифікацію й огляд СКУД, та найбільш поширений аналог, можна створити принципову схему проекту, відокремити блоки, підібрати протоколи зв'язку, за якими вони будуть взаємодіяти.

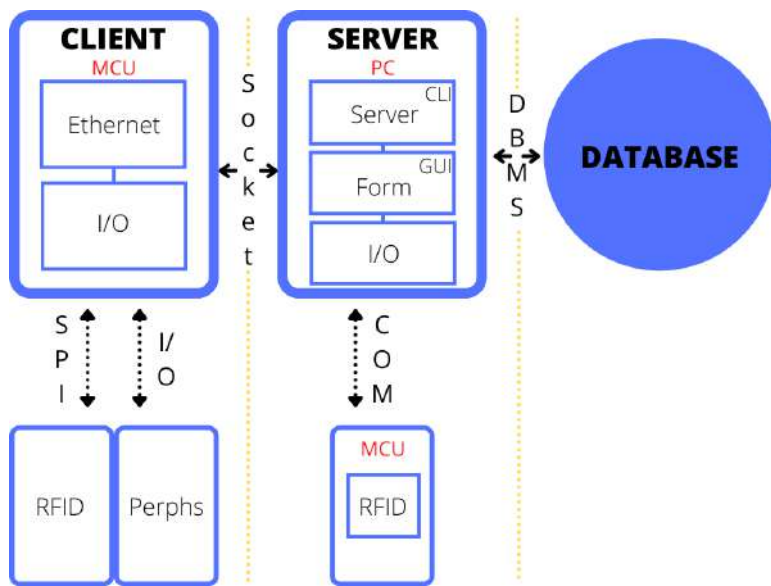


Рисунок 1.11 – Принципова схема системи

Структура системи контролю та управління доступом запозичує основу клієнт-серверної топології будовання подібних комплексів (рисунок 1.9). Як було вказано вище, принцип побудови такого зразка надає більш широкий функціонал під час роботи та має вищий рівень безпеки зберігання та опрацювання даних.

Система має керуватись із спеціального пункту контролю, який має знаходитись на деякому віддаленні від основного «шлюзу», де реалізується фізичний пропуск. За допомогою спеціального пункту контролю мають бути реалізовані такі можливості як: зберігання даних користувачів та їх унікальних ідентифікаційних номерів, безперервний контроль та фіксація дати та часу подій входу та виходу користувачів, внесення, зміна та видалення даних користувачів.

СКУД має відрізнятись властивостями до боротьби із електромагнітними, електричними та несанаційними програмними намаганнями втрутитись у її робочий процес. Також, має бути витриманий баланс між бюджетними прототипами локальних систем контролю та професійними реалізаціями клієнт-серверних систем, які працюють на підприємствах та у деяких навчальних закладах, тощо.

Отже, система контролю та управління доступом належить до систем мережевого клієнт-серверного типу, з одним пунктом пропуску, але з апаратною та програмною можливістю розширювати комплекс, з'єднуючи в масив. Спосіб ідентифікації- радіочастотний з унікальним номером ідентифікації «UID». Оскільки прототип є тестовим та проект із програмним та електронно-апаратним поглибленням- механічна система чітко визначена не була і залишається із можливістю вибору між замками електромагнітного типу та турнікетними типами.

## 1.5 Висновки за розділом

Було здійснено огляд предметної області охоронних та контролюючих систем СКУД, їх класифікації. Також було оглянуто найбільш популярне рішення від надійного виробника, описано сферу використання технології радіочастотної ідентифікації RFID та її вагу під час використання у реальних умовах. Створено принципову схему СКУД до проектування залежно від потреб до системи та обрано напрямок вибору базових вузлів.

## РОЗДІЛ 2

### ВИБІР ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ПРОЕКТУ

2.1 Вибір та обґрунтування базових вузлів для створення апаратного комплексу проекту

Система контролю та управління доступом- це складний механізм, який складається із сучасних обчислювальних блоків, де кожен з них чітко виконує своє встановлене завдання. Так званий «ринок» високотехнологічних пристроїв відкриває великий простір для втілення різноманітних інженерних думок та алгоритмів. Від якості реалізації схемотехніки та сумісності обраних електронних компонентів залежить безвідрадість роботи всієї системи в цілому.

На момент написання роботи основою або базою будь-якого електронного пристрою, в незалежності від того програмується він після цього чи ні, є напівпровідникові інтегральні мікросхеми (ІС). В цьому розділі дипломної роботи описано алгоритм вибору компонентної бази, обґрунтовано вибір кожного вузла об'єднаної системи та створено принципову схему виконавчого пристрою (клієнту в клієнт-серверній СКУД) із проектуванням та виробництвом друкованої плати з дотриманням основних вимог інженерного проектування.

#### 2.1.1 Контролер виконавчого пристрою

Клієнт-серверна система контролю та управління доступом в частині виконавчого пристрою вимагає використовувати центральний обчислювальний блок. Майже в кожному сучасному пристрої, починаючи від інфрачервоного пульта управління або холодильнику, закінчуючи системами управління нахилу двигунами космічної ракети встановлено програмовану інтегральну мікросхему. Називаються такі мікросхеми мікроконтролерами або

мікропроцесорами. Приставка «мікро» характеризує їх щільність розташування транзисторних елементів на кристалі кремнію.

Традиційно, мікропроцесор (MPU-Micro Processor Unit)- це пристрій, який відповідає за виконання арифметичних та логічних операцій та операцій керування, які записані у машинному коді. Реалізований він на основі одного або двох кристалів кремнію. Мікропроцесор не обов'язково потребує додаткових пристроїв для його роботи, які зазвичай підключаються зовні, а саме: постійно-запам'ятовуючий та оперативно-запам'ятовуючий пристрої. В сучасності завдяки рідкісному використанню процесорів в реальному житті, які не є мікропроцесорами, в побутовій лексиці терміни «мікропроцесор» та «процесор» рівнозначні.

Внутрішня побудова мікропроцесора: арифметико-логічний пристрій (АЛП), блок керування та синхронізації, запам'ятовуючий пристрій, регістри, шини передачі даних та команд.

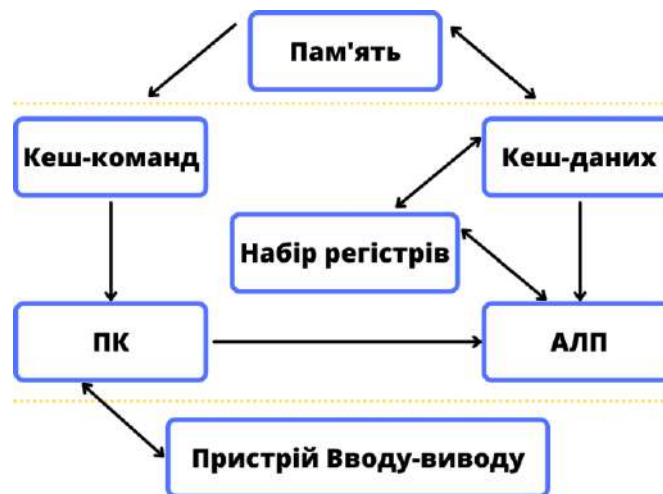


Рисунок 2.1 – Зображення схематичної побудови мікропроцесора

Особливістю мікропроцесорів є велика потужність обробки даних. Вони створюються для використання в потужних системах, таких як: сервери, персональні комп'ютери, телефони та інші спеціалізовані прилади, де потрібно обробляти великий об'єм інформації.

Мікроконтролер (MCU- Micro Controller Unit)- інтегральна мікросхема, що побудована на кристалі кремнію, так само, як і мікропроцесор. Типічний



мікроконтролер об'єднує на кристалі мікропроцесор, периферійні пристрої, що потрібні для роботи мікропроцесора (ПЗП, ОЗП) та додаткові прилади для роботи із зовнішніми пристроями, такі як: АЦП (Аналого-Цифровий Перетворювач), ЦАП (Цифро-Аналоговий Перетворювач), таймери різного призначення. Мікроконтролери ще називають одно кристалними комп'ютерами, через те, що вони не потребують майже ніяких додаткових блоків та є працездатними самі.

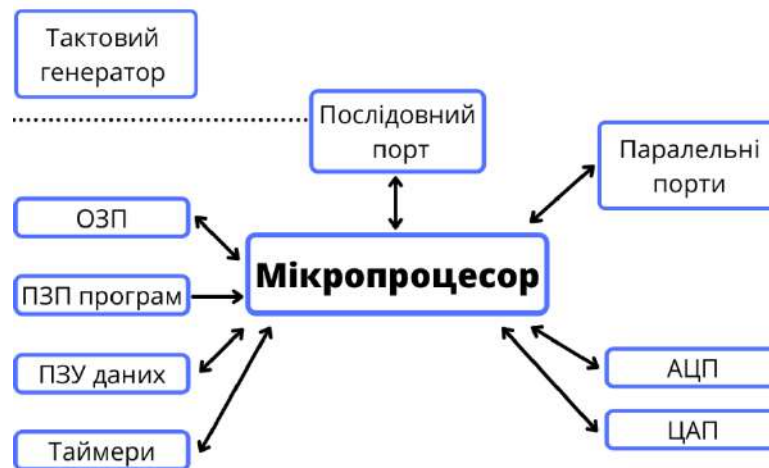


Рисунок 2.2 – Зображення схематичної побудови мікроконтролера

Мікроконтролер може виконувати великий спектр нескладних завдань, на відміну від мікропроцесора: керування периферійними пристроями за допомогою відкритих протоколів, керування силовими електромагнітними та індуктивними механізмами, за допомогою силових напівпровідників, керування дисплеями, блоками керування та інші. Зона відповідальності мікроконтролерів- керування вбудованих та переносних пристроїв. Також, мікропроцесор є сумісним в роботі з мікроконтролером та зазвичай застосовується в зв'язці.

Таблиця 2.1 – Порівняльна характеристика мікроконтролера та мікропроцесора

	<b>Мікропроцесор</b>	<b>Мікроконтролер</b>
Реалізація	Складається з однієї потужної інтегрованої мікросхеми та потребує зовнішніх пристроїв для функціонування	Складається з мікропроцесора невеликої потужності та містить усі необхідні для функціонування блоки
Характеристика	Залежна одиниця	Автономний блок
Порти вводу/виводу	-	+
Цільове призначення	Ринок найвищого класу	Вбудовані системи
Енергоспоживання	Не має вбудованого контролера розподілення живлення та потребує зовнішнього	Має вбудований контролер живлення та менш прискіпливий до якості живлення

Отже, проаналізувавши цільові напрямки використання двох видів сучасних обчислювальних пристроїв зроблено висновок, що в проекті системи контролю та управління доступом складських приміщень доцільніше використати основою обчислювального пристрою мікроконтролер. Система не потребує вирішувати тяжкі обчислювальні процеси і, навпаки, потребує деякої кількості вбудованих портів вводу та виводу інформації, роботи з протоколом з'єднання, таким як SPI (Serial Pheriphiral Interface).

Щодо різноманіття мікроконтролерів, то тут вибір ще більший. Існують безліч сучасних одно кристальних електронно-обчислювальних машин: Motorola, Zilog, AVR від Atmel, Nordic, ESP від Tensilica, PIC від Microchip, STM від STMicroelectronics та інші. Велика їх кількість здатна опрацьовувати

потреби СКУД складських приміщень, тому треба вирішити який саме вибрати.

На момент написання дипломної роботи найбільшу популярність серед професійних розробників вбудованих систем мають сучасні мікроконтролери від компаній Microchip та STMicroelectronics, серії PIC та STM32 відповідно. До створення проекту була нагода працювати саме с мікроконтролерами STM32. Вони мають сучасну архітектуру, найвищі показники за характеристиками, чудове співвідношення ціни та якості та якісні компілятори від виробника мікроконтролерів. Компанія STMicroelectronics працює з багатьма відомими компаніями та структурами у авіаційній та космічній промисловості, тож за якість можна не хвилюватись.

Було обрано будувати систему на базі мікроконтролеру від цього виробника.



Рисунок 2.3 – Зображення торгівельного знаку компанії ST

Мікроконтролери виробника STMicroelectronics поділяються на декілька сімейств за розрядністю мікропроцесора, а саме: STM8 (8-розрядний), STM16 (16-розрядний), STM32 (32-розрядний). Було обрано 32-розрядну версію. По-перше, через те, що швидкість обробки даних у такого контролера набагато більша, що дозволяє працювати з високошвидкісними протоколами передачі даних. По-друге, вартість 32-розрядної версії майже така ж, як і у інших.

Основою реалізації було обрано STM32F103. Потужний мікроконтролер із обчислювальним ядром мікропроцесора Cortex-M3 з частотою обчислювання ядра та головної шини 72МГц. Також, присутня велика

кількість портів вводу-виводу (I/O ports) та 2 окремих незалежних SPI (Serial Peripheral Interface) протоколи.

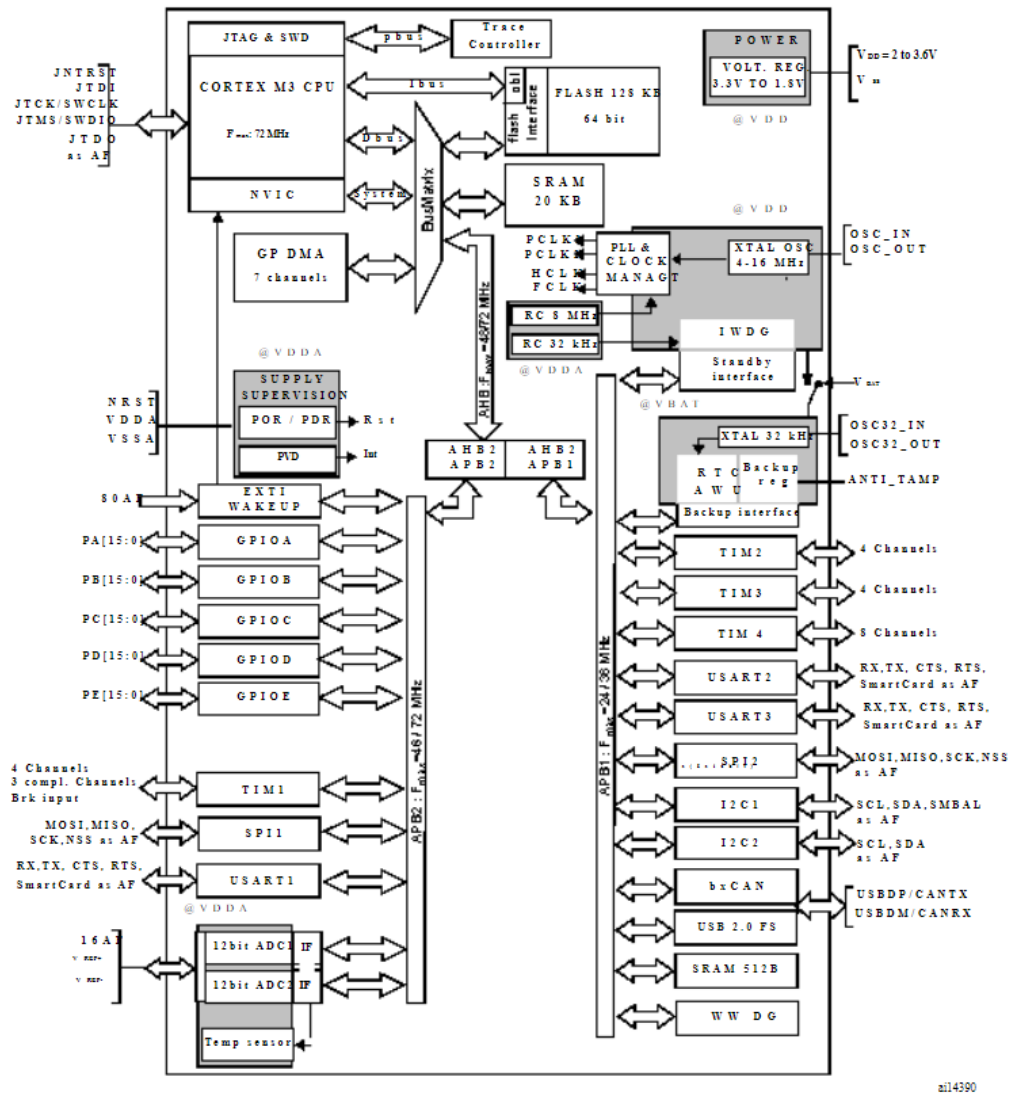


Рисунок 2.4 – Зображення блочної побудови мікроконтролера STM32F103

Таблиця 2.2 – Характеристика мікроконтролера STM32F103

	Мікроконтролер STM32F103
Flash пам'ять	64 КБ
SRAM пам'ять	20 КБ
Базові таймери	23 одиниці
Спеціальні таймери	11 одиниць
Протоколи SPI	2 одиниці
Протоколи I2C	2 одиниці

Протоколи USART	3 одиниці
Протоколи USB	1 одиниця
Протоколи CAN	1 одиниця
Шини входу та виходу GPIO	32 одиниці
12-розрядні АЦП	2 таймери по 10 каналів
Частота тактування ядра	72 МГц
Напруга живлення	Від 2 до 3,6 вольт
Температурний діапазон	Від -40 до +80 (за Цельсієм)

### 2.1.2 Контролер радіочастотної ідентифікації

Пристрій безконтактного обміну цифрових даних виконує функцію передавача та приймача даних до «смарт-карти» або електронного ключа та прийому записаних даних в внутрішню пам'ять від них. Контролер на його спорядження є одним із центральних вузлів, які відповідають за безпеку та якість роботи усієї системи та того, що вона охороняє. Річ іде саме про контролер, що керує антеною передавача поблизу основного контролера «клієнта» в клієнт-серверній системі.

Оскільки раніше було зазначено, що обрано використовувати карти або ключі за технологією Mifare від NXP, то треба звернути увагу на якій частоті обміну вони працюють. Це 13,56 МГц, отже й передавач обрано саме за цим принципом.

Основною перевагою є те, що прийому-передавач MRFC522 випускається тією ж самою компанією Nezipria, отже саме для роботи за технологією Mifare. Мікросхема має велику та чітко сформульовану офіційну документацію для створення потрібних програмних бібліотек та функцій, а також для апаратного підключення.

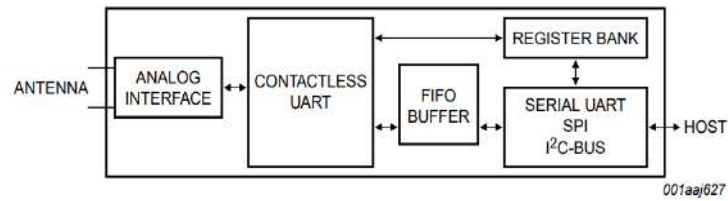


Рисунок 2.5 – Зображення структурної побудови контролера радіочастотної ідентифікації MFRC522

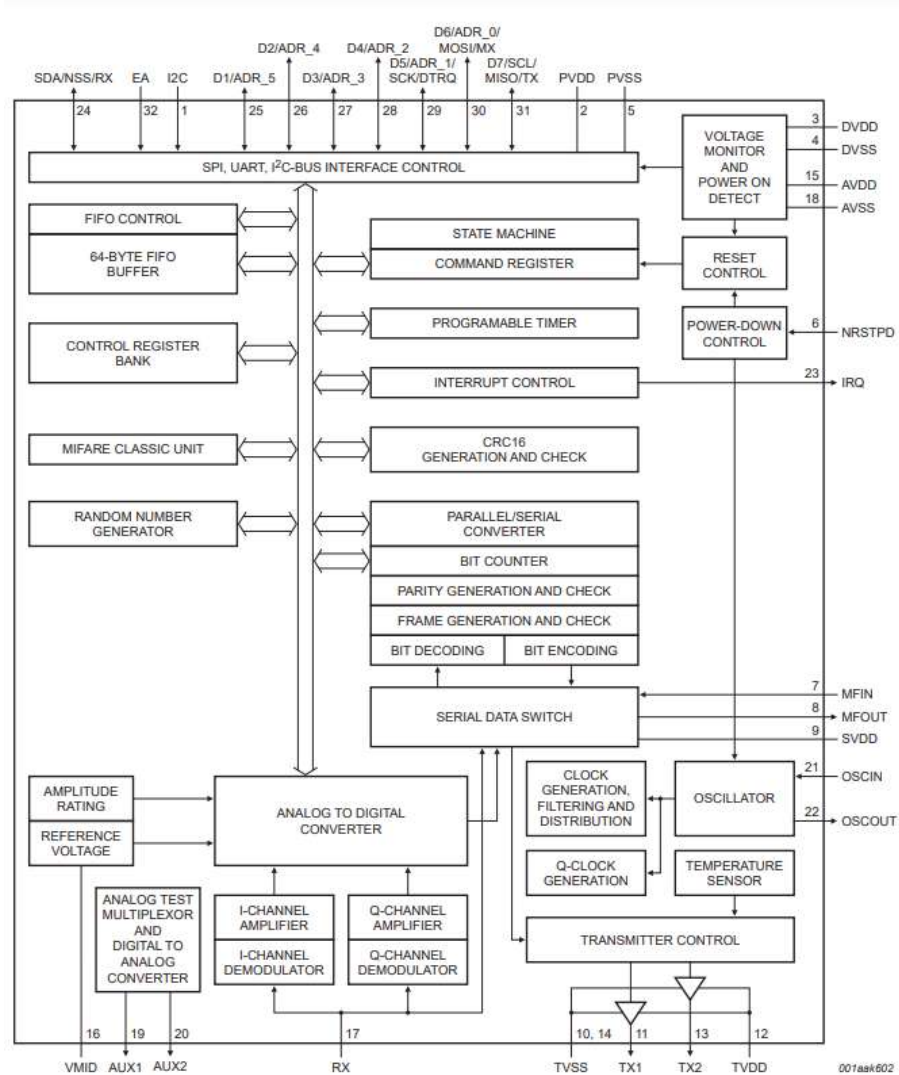


Рисунок 2.6 – Зображення блочної побудови контролера радіочастотної ідентифікації MFRC522

Мікросхема MFRC522 може керуватися за різними основними протоколами зв'язку, а саме: SPI, I2C та UART. Живиться від тієї ж напруги, що і основний контролер.

До мікросхеми під'єднується «антена», а якщо бути точніше, то електромагнітна індуктивна котушка із чотирьох витків проводу. Вона одночасно виконує функцію живлення для «смарт-карт» транспондерів та прийому-передачі інформації. За сутністю- це трансформатор, який не має магнітного сердечника.

Протоколом передачі обрано SPI (Serial Peripheral Protocol). Він найшвидший із запропонованих та здатен забезпечувати швидкість передачі інформації до 10 Мб/сек. В проекті СКУД складських приміщень обмін інформації відбувається на швидкості 6 Мб/с для запобігання від втручання електромагнітних перешкод на дуже великій швидкості передачі. Наприклад, швидкість I2C у даному випадку всього 100 або 400 Кб/с у швидкому режимі а UART до 1228.8 Мб/с.

Таблиця 2.3 – Характеристика контролера MFRC522

	<b>MFRC522</b>
Напруга живлення	Від 2,5 до 3,3 вольт
Протоколи комунікації	SPI, I2C, RS232 UART
Швидкість SPI	До 10 Мб/секунду
Частота тактування ядра	27,12 МГц
Сопроцесор підрахунку контрольної суми (CRC)	+
Дальність зчитування міток	До 50 міліметрів від антени до антени
Підтримка карток	A/MIFARE за стандартом ISO/IEC14443
Вбудований тест самоперевірки	+

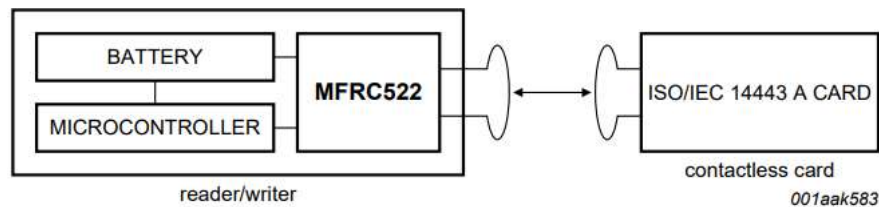


Рисунок 2.7 – Зображення методу взаємодії радіочастотного контролера із транспондером

### 2.1.3 Чіп зберігання ідентифікаційних даних

Щодо логіки роботи мікросхеми-транспондеру Mifare від NXP, або мікросхеми, яка зберігає дані було частино розглянуто раніше. Обраний Mifare версії Classic- це сімейство, що складається з карт: MIFARE Classic 1K, MIFARE Classic 4K, MIFARE Classic EV1 1K, MIFARE Classic EV1 4K, MIFARE ID и MIFARE Mini.

Mifare Classic 1K пропонують надбудову за стандартом ISO 14443A-3 з криптографічним захистом даних. Містить 4-х байтні або 7-ми байтні незмінювані унікальні ідентифікаційні номери карт та 1KB або 4KB даних користувача та конфігураційних даних. Внутрішня пам'ять типу EEPROM, що є незалежною від живлення.

У картах Mifare Classic використовується пропрієтарний ліцензійний криптоалгоритм Crypto-1. Спочатку стійкість алгоритму була заснована на його таємності. Алгоритм не розголошувався, використовувати його можна було лише у складі мікросхем Philips (пізніше NXP Semiconductors). Однак низька криптостійкість алгоритму і популярність технології призвела до того, що на сьогоднішній день алгоритм не є секретом і легко зламується.

Але в реальних системах далеко не вся безпека побудована на апаратному шифруванні карти. Як додатковий фактор захисту можуть використовуватися, наприклад, мітки часу. Тим не менш, навіть системи, безпека яких не спирається повністю на алгоритм Crypto-1 (або навіть не використовує його зовсім як Mifare Ultralight), можуть бути зламані завдяки апаратним особливостям карт.



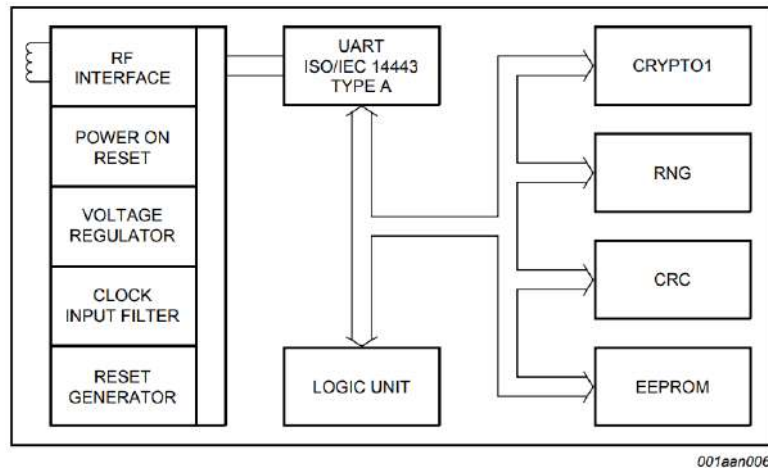


Рисунок 2.8 – Зображення блочної побудови інтегральної мікросхеми зберігання ідентифікаційних даних

Для карт стандарту 1K перші 1024 байт організовані у 16 секторів по 64 байти. Кожен сектор поділений на 4 блоки. Останній блок кожного сектора називається трейлером сектора, у ньому записані ключі та параметри доступу до сектора. Нульовий блок нульового сектора – це спеціальний блок, заблокований на запис, він містить ідентифікатор та інформацію виробника картки.

Перед зверненням до сектора для читання або запису необхідно виконати авторизацію за допомогою ключа розміром 6 байт. Авторизація відбувається за три етапним протоколом, при цьому використовується пропрієтарний алгоритм потокового шифрування Crypto1.

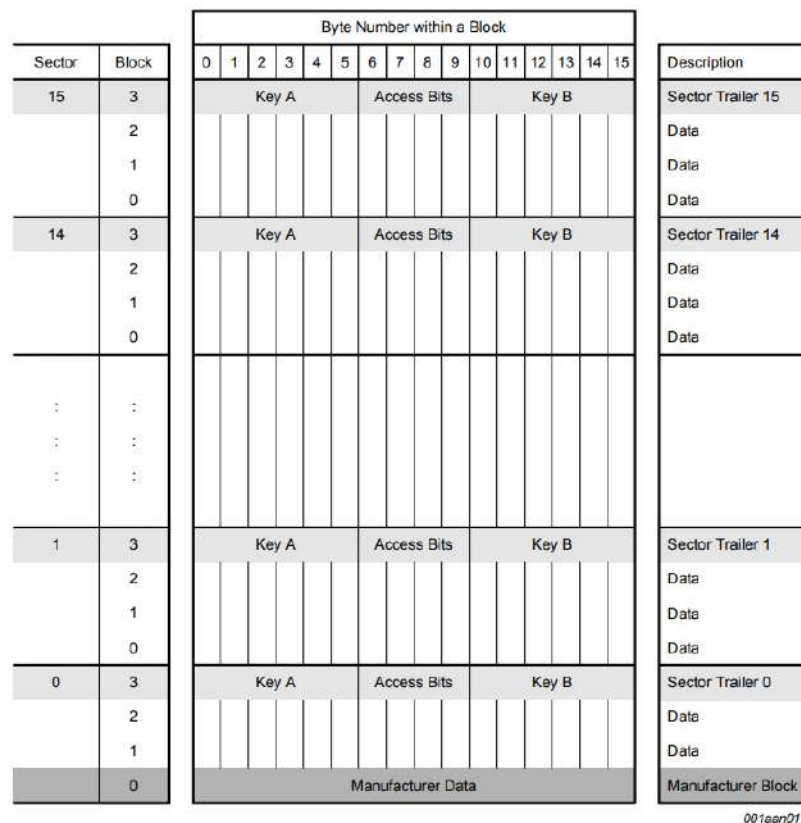


Рисунок 2.9 – Зображення структурної схеми кластерів зберігання даних мікросхеми зберігання даних

Комунікативний процес або процес з'єднання із зчитувачем відбувається за наступним алгоритмом, що описаний у технічній документації виробника електронних ключів Mifare. Команди ініціюються зчитувачем і контролюються цифровим блоком керування MF1S50ууX/V1.

1) Цикл стандартного запиту.

Після того, як на висновок перезавантаження або Reset (POR-Power On Reset) подається живлення карта відповідає на запит команду з кодом відповіді на запит.

2) Цикл анти колізії (боротьби із зіткненням).

У циклі анти колізії зчитується ідентифікатор карти. Якщо є кілька карток в операційному полі зчитувача, то їх можна розрізнити за ідентифікатором і може бути вибрана тільки одна карта для здійснення подальших операцій. Невибрані карти повертаються в режим очікування стану і чекають нової команди запиту.

### 3) Цикл вибору карти.

За допомогою команди блок зчитування обирає одну окрему картку для автентифікації і операції, пов'язані з пам'яттю та картка повертає ідентифікаційний унікальний номер.

### 4) Цикл три факторної автентифікації.

Після вибору карти блок зчитування вказує розташування доступу до пам'яті та використовує відповідний ключ для три прохідної автентифікаційної процедури. Після успішної автентифікації всі команди та відповіді шифруються.

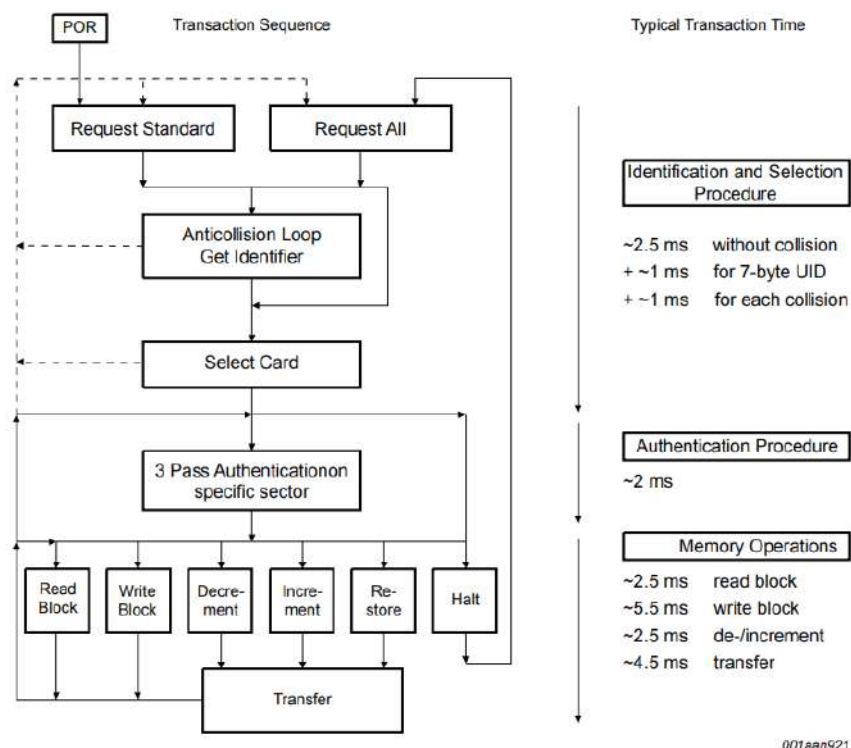


Рисунок 2.10 – Зображення алгоритму обробки даних контролером радіочастотної ідентифікації MFRC522

Після проходження процесу автентифікації відкриваються наступні можливості роботи із пам'яттю EEPROM:

- 1) Читання одного із 16 блоків
- 2) Запис в один із 16 блоків
- 3) Декримент: зменшує вміст блоку та зберігає результат у внутрішньому буфері

- 4) Інкремент: збільшує вміст блоку та зберігає результат у внутрішньому буфері
- 5) Відновлення: переміщує вміст одного з 16 блоків у внутрішній буфер передачі
- 6) Передача: записує вміст внутрішнього буфера передачі в блок значення

Таблиця 2.4 – Характеристика транспондера Mifare Classic EV1 1K

	<b>Mifare Classic EV1 1K</b>
Безконтактна передача даних та живлення	+
Робоча частота	13,56 МГц
Сопроцесор підрахунку контрольної суми (CRC)	16-розрядний сопроцесор контрольної суми
Час виконання однієї операції	Менше 100 мс
Дальність зчитування	Менше 50 мм
Швидкість передавача	106 Кб/с
Механізм антиколізії	+
Довжина ідентифікаційного номеру	4 байти або 7 байтів
Тип вбудованої пам'яті	EEPROM
Об'єм вбудованої пам'яті	1024 байти (16 секторів по 64 байти)
Кількість циклів перезапису	Біля 200 000

#### 2.1.4 Контролер з'єднання мережі

В будь-яких системах клієнт-серверного типу обов'язково є відокремлена частина «клієнт», де відбуваються одні обчислювальні процеси та відокремлена частина «сервер», в якій аналогічно проходять обчислення спираючись на завдання. Ці дві системи хоч і є відокремленими, але все одно працюють у тандемі, виконуючи глобальне завдання всієї системи.

Клієнт-серверна система контролю та управління доступом складських приміщень не є виключенням. Судячи з самої назви, система побудована на двох відокремлених вузлах обміну інформацією.

З персональним комп'ютером, на основі якого побудований сервер, все зрозуміло. В ньому вбудована мережева карта, за допомогою якої можна під'єднуватись до будь-якого вузла мережевого зв'язку, якщо є його IP (Internet Protocol)- номер. Апаратну частину мережевої карти зазвичай не довго й вибирають. Сучасні мережеві карти мають гарні характеристики, тим більш для проекту такого рівня велика швидкість та складність обміну не потрібна, а у половині випадків, наприклад у ноутбуках, вона вмонтована на головну друковану плату, що ще більш спрощує процес вибору та подальшого проектування системи.

З «клієнтом», що у даному випадку є виконавчий блок клієнт-серверної системи трохи складніше. Через те, що проектування вбудованих систем займає більш специфічну та відокремлену нішу, аніж збірка персонального комп'ютеру з готових фабричних рішень, воно потребує більш щільного аналізу для подальшої інтеграції в систему.

Як було вказано раніше (див. 1.2.1 Принципова схема СКУД складського приміщення), СКУД тримає зв'язок за протоколом пакетної передачі даних Ethernet (TCP/IP), то й контролер потрібен із його підтримкою.

Проаналізувавши технологічний «ринок», який пропонує контролери Ethernet-зв'язку для проектування та використання у вбудованих системах (IoT- Internet Of Things) було з'ясовано, що однією з найпопулярніших виробників мікросхем класу TCP/IP є компанія WIZnet.

WIZnet - це компанія, що спеціалізується за платформою інтернету речей. Виробник обіцяє технологію жорсткого «TCP/IP», яка забезпечує більш стабільну та продуктивну роботу «вбудованого приладу», в порівнянні з використанням технологій програмного рішення локального або глобального зв'язку на базі основного обчислювального контролеру системи. Якщо пояснювати більш спрощено- контролер WIZnet безпосередньо бере участь у

роботі та виконує усі складні алгоритми обчислень Ethernet своїми силами. Це значно покращує становище для мікроконтролера STM32F103.



Рисунок 2.11 – Зображення торговельного знаку компанії WIZnet

Компанія WIZnet пропонує кілька апаратних ітерацій контролерів зв'язку, а саме: W3150A+, W6100, W5100S, W5100, W5200, W5300, W5500. Через те, що так званім «флагманом» є 5500 версія, було обрано саме її.

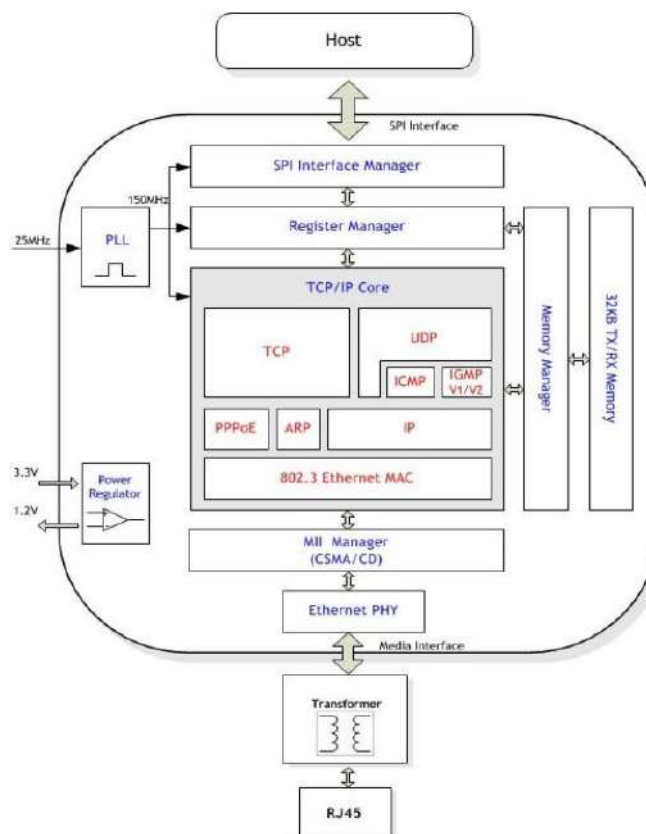


Рисунок 2.12 – Зображення схеми блоку мережевої комунікації Ethernet

WIZnet W5500- це вбудований Ethernet-контролер з підключенням TCP/IP, який забезпечує спрощене підключення до інтернету або локальної мережі для вбудованих систем за допомогою SPI (Serial Peripheral Interface).

W5500 найкраще підходить у випадках, коли потребується стабільне підключення до інтернету або локальної мережі, використовуючи єдиний чіп

для реалізації стека TCP/IP, 10/100 Ethernet MAC та PHY. Стек TCP/IP підтримує TCP, UDP, IPv4, ICMP, ARP, IGMP і PPPoE та інші. Це було доведено в різних програмах протягом багатьох років. W5500 використовує внутрішній буфер 32 Кбайт як свою пам'ять для передачі даних.

Використовуючи W5500, можна реалізувати програму Ethernet, використовуючи технологію простих програмних сокетів замість роботи зі складним контролером Ethernet, з потребою написання складних алгоритмів. Можливе використання 8 незалежних апаратних сокетів одночасно.

SPI (Serial Peripheral Interface) забезпечує легку інтеграцію із зовнішнім мікроконтролером. W5500 SPI підтримує швидкість передачі даних до 80 МГц. У випадку потреби зменшити енергоспоживання системи, W5500 забезпечує WOL (Wake on LAN) і режим вимкнення живлення.

Таблиця 2.5 – Характеристика контролера WIZnet W5500

	<b>WIZnet W5500</b>
Протоколи взаємодії	TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
Кількість сокетів	8 незалежних сокетів
Режим вимкнення живлення	+
Режим запуску за зовнішнім сигналом	+
Інтерфейс зовнішнього зв'язку	Високошвидкісний SPI (до 80 МГц)
Внутрішня пам'ять	32 КБ для буферів TX/RX
Режими роботи	Повний і напів-дуплекс
Фрагментація IP	-
Напруга живлення	3,3 В
Толерантність до напруги 5 вольт логічних сигналів	+
Зовнішня індикація	+

За основну швидкість роботи зовнішнього інтерфейсу комунікації SPI обрано 18 МГц. Мікросхема контролера локальної мережі будується біля основного мікроконтролера, що дозволяє працювати на високій швидкості. За конструктивними характеристиками тактування основного мікроконтролера STM32F103 швидкість роботи шини оптимальна.

Окрему увагу можна приділити якісній програмній бібліотеці від виробника WIZnet. Вона має конструкти усіх базових програмних функцій для роботи з контролером, що спрощує написання програми.

#### 2.1.5 Контролер пристрою внесення даних

Пристрій внесення даних- це електронний прилад, завдяки якому створюється внесення та перезапис ідентифікаційної інформації, що знаходиться у таблицях бази даних, яка в свою чергу зберігає дані користувачів, що потрібні для коректної роботи із системою контролю та управління доступом складських приміщень.

Було вирішено створення приладу на базі мікроконтролера, через те, що від процесу обробки такої інформації не потребується потужність системи та велика кількість портів для взаємодії.

В цьому випадку використовується 8-розрядний одно кристальний комп'ютер від компанії Atmel сімейства Mega моделі 328. На момент написання дипломної роботи він є найбільш популярним серед інших моделей цієї компанії та є повністю відповідаючим вимогам проекту СКУД. Також, його вартість є найменшою серед інших мікроконтролерів від Atmel, навіть, не дивлячись на його кращі характеристики за деякі моделі та серії.



Рисунок 2.13 - Зображення торгівельного знаку компанії Atmel



Апаратно, мікроконтролер пристрою внесення даних зв'язує другий контролер радіочастотної ідентифікації із персональним комп'ютером, на якому базується серверна частина проекту. Як і у випадку з вузлом «клієнта», MFRC522 комунікує з мікроконтролером задіявши протокол зв'язку SPI.

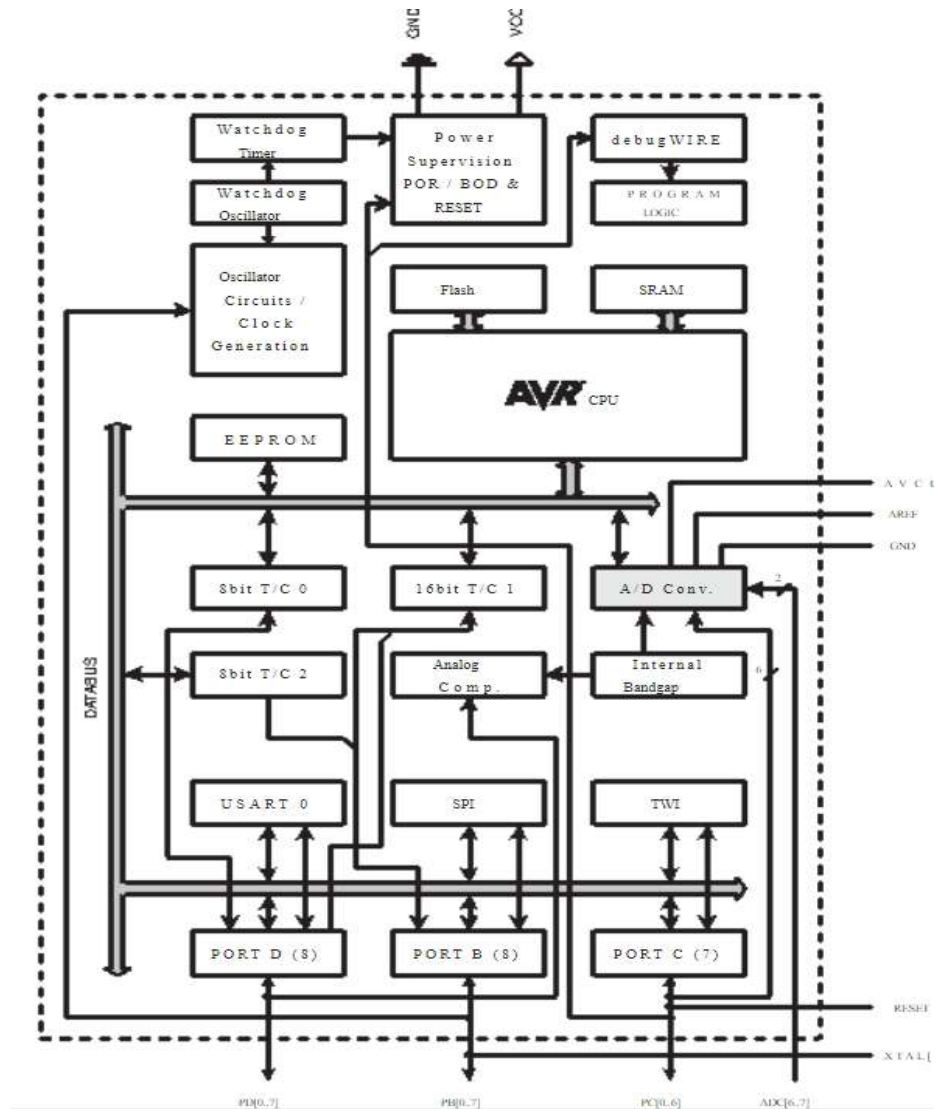


Рисунок 2.14 – Зображення структурної побудови мікроконтролера ATmega 328

Основою ж з'єднання мікроконтролера з персональним комп'ютером є мікросхема-перетворювач серійного зв'язку SERIAL до універсальної синхронної шини USB. За допомогою цього рішення пристрій є переносним, через використання відомого стандарту зв'язку, апаратна можливість якого надається більшістю персональних комп'ютерних систем.

Таблиця 2.6 – Характеристика мікроконтролера ATmega 328

	<b>Мікроконтролер ATmega 328</b>
Flash пам'ять	32 КБ
EEPROM пам'ять	512 Б
SRAM пам'ять	2 КБ
Базові таймери	2 одиниці (8 розрядні)
Спеціальні таймери	1 одиниць (16 розрядний)
Протоколи SPI	1 одиниця
Протоколи I2C	1 одиниця
Протоколи USART	1 одиниця
Протоколи USB	1 одиниця
Протоколи CAN	-
Шини входу та виходу GPIO	23 одиниці
10-розрядні АЦП	14 каналів
Частота тактування ядра	16 МГц
Напруга живлення	Від 1,8 до 5,5 вольт
Температурний діапазон	Від -40 до +85 (за Цельсієм)

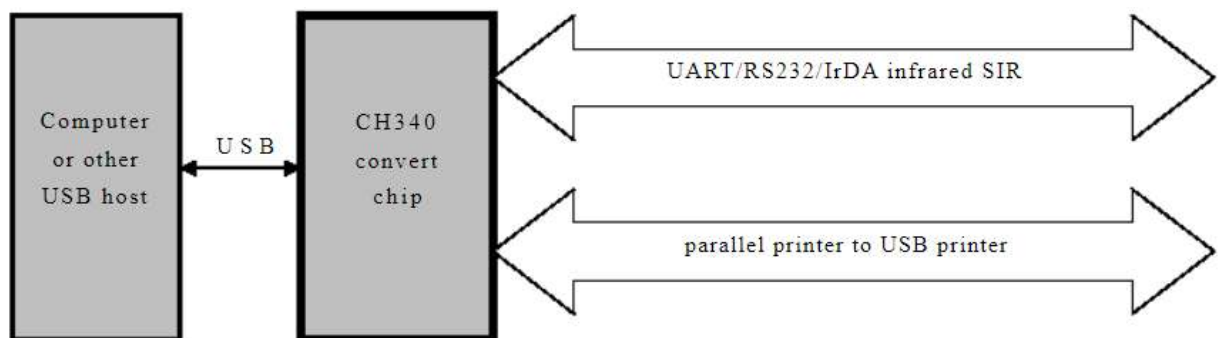


Рисунок 2.15 – Зображення методу підключення мікросхеми-перетворювача із комп'ютерної системою

Таблиця 2.7 – Характеристика мікросхеми СН 340

	<b>СН 340</b>
Підтримка USB	+, версії 2.0
Підтримка операційних систем зовнішнього хосту	Windows та інші
Швидкість передавача	Від 50 б/с до 2 Мб/с
Режими роботи	Повний та напів-дуплекс
Толерантність до різних логічних рівнів	+, 3,3 та 5 В
Напруга живлення	Від 3,3 до 5 В

## 2.2 Вибір та обґрунтування базових рішень для створення програмного комплексу проекту

Створення програмного комплексу системи контролю та управління доступом складських приміщень- друга проектна частина. Від коректного підбору сучасного та якісного програмного забезпечення також залежить безвідрадність та якість роботи всієї системи.

Структура проектної частини було вирішено складати з чотирьох програмних кодів, за допомогою яких можна користуватися функціоналом СКУД.

Перша частина- програмне забезпечення для програмування блоку виконавчого пристрою або «клієнта».

Друга частина- програмне забезпечення для програмування блоку серверної частини або «сервера».

Третя частина- програмне забезпечення для програмування блоку внесення та корегування даних користувачів.

Четверта частина- програмне забезпечення для задля програмування графічного додатку внесення ідентифікаційних номерів, що встановлюється на персональний комп'ютер.

У цьому розділі дипломної роботи аналізується різноманіття програмного забезпечення. Спираючись на це, робиться висновок щодо програмного забезпечення, що буде використовуватись в роботі. Також, обґрунтовується вибір окремих рішень.

### 2.2.1 Програмне забезпечення для програмування блоку виконавчого пристрою

Як було описано раніше, апаратна частина блоку виконавчого пристрою базується на одно кристальному ЕОП (Електронно-обчислювальному пристрої) STM32 моделі F103 від компанії виробника STMicroelectronics. За роки професійної роботи виробник зміг створити багато різноманітних програмних продуктів, спрямованих на програмування усієї лінійки мікроконтролерів STM, яких у неї більше 16 повноцінних серій.

Однак, існує інший продукт, який має велику популярність серед програмних інженерів та не є плодом розробки від ST.

Отже, порівнюється 2 основних середовища розробки для ведення розробки програмної частини «клієнта»: CubeIDE та KEIL uVision. Також, до інтегрованого середовища розробки CubeIDE належить програмний додаток, що візуалізує та спрощує процес конфігурації налаштувань, через це можна вважати, що додається ще й третій пункт порівняння Cube MX.

KEIL uVision- програмний продукт, що дозволяє працювати з проектами будь-якого ступеня складності, починаючи з введення та виправлення вихідних текстів і закінчуючи внутрішньо схемним налагодженням коду та програмуванням ПЗП мікроконтролера. Від розробника прихована більшість другорядних функцій, що сильно розвантажує інтерфейс і робить управління інтуїтивно зрозумілим.

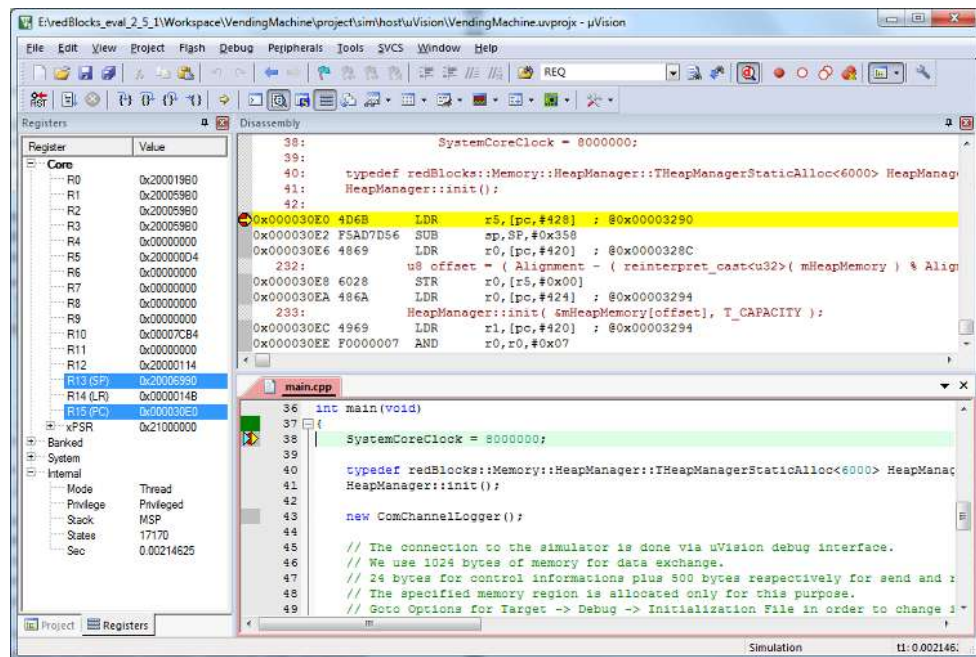


Рисунок 2.16 – Зображення робочого вікна програми «Keil»

KEIL uVision користується багато інженерів програмного забезпечення, середовище не нове, спочатку було розроблено аж у кінці минулого століття. Програмне забезпечення вважається гарно виваженим та допрацьованим, інтерфейс якого інтуїтивно зрозумілий. Також, за середовищем написано багато статей та є велика кількість документації.

Таблиця 2.8 – Порівняння програми KEIL uVision

	<b>KEIL uVision</b>
Переваги	Недоліки
Велика база до мікроконтролерів, де є докладна інформація до всіх пристроїв	Інколи невірні вказівники помилок синтаксису коду
Менеджер проектів	Комерційний продукт з великою вартістю авторизації
Вбудований редактор коду, що полегшує роботу з вихідним текстом	Мала пам'ять програми, що компілюється, у випадку

	використання безоплатної версії (до 32 КБ)
Відладчик-симулятор, що допомагає налагоджувати роботу програми, що була скомпільована	Майже відсутня довгострокова підтримка проекту для подальшого керування, корегування та вдосконалення. Через деякий період проект можна не відновити
Проста побудова інтерфейсу інтегрованого середовища розробки	Відсутня підтримка інших операційних систем окрім Windows
Велика кількість інформації та документації щодо роботи з середовищем розробки	Доволі не новий продукт, що не мав великих вдосконалення за останні часи експлуатації

Висновок: проаналізувавши характеристики, властивості, переваги та недоліки інтегрованого середовища розробки KEIL uVision було з'ясовано, що більш за все, воно підходить для використання на великих підприємствах для великих компаній, де йдеться постійна розробка профільних продуктів у сфері вбудованої електроніки та тих компаній, де програмне забезпечення комп'ютерних систем не змінюється довгостроково.

Інтегроване середовище розробки Cube IDE та візуальний конфігуратор Cube MX буде розглянуто сумісно, адже вони позиціонуються як єдиний продукт. Тим не менш, це 2 різних програмних продукти.

Cube IDE- це відносно нове середовище, що було анонсоване у 2019 році компанією STMicroelectronics як спеціалізований продукт програмної розробки під мікроконтролери тієї в компанії. Програмний продукт є безоплатним як для комерційного, так і для приватного використання. Він започаткував усі відносно старі та базові функції інших, більш старих середовищ розробки для мікроконтролерів.

Програмування ведеться мовою C, з використанням базових бібліотеки HAL та бібліотек низького рівня CMSIS.

Бібліотека HAL (Hardware abstraction layer) має в своєму розпорядженні функції налаштовані на високий рівень програмування (інтуїтивно зрозуміли). Використовується у більшій частині програмного коду через легкість використання.

Structure field	Description
Pin	Specifies the GPIO pins to be configured. Possible values: GPIO_PIN_x or GPIO_PIN_All, where x[0..15]
Mode	Specifies the operating mode for the selected pins: GPIO mode or EXTI mode. Possible values are: <ul style="list-style-type: none"> <li>• <u>GPIO mode</u> <ul style="list-style-type: none"> <li>– GPIO_MODE_INPUT : Input floating</li> <li>– GPIO_MODE_OUTPUT_PP : Output push-pull</li> <li>– GPIO_MODE_OUTPUT_OD : Output open drain</li> <li>– GPIO_MODE_AF_PP : Alternate function push-pull</li> <li>– GPIO_MODE_AF_OD : Alternate function open drain</li> <li>– GPIO_MODE_ANALOG : Analog mode</li> </ul> </li> <li>• <u>External Interrupt mode</u> <ul style="list-style-type: none"> <li>– GPIO_MODE_IT_RISING : Rising edge trigger detection</li> <li>– GPIO_MODE_IT_FALLING : Falling edge trigger detection</li> <li>– GPIO_MODE_IT_RISING_FALLING : Rising/Falling edge trigger detection</li> </ul> </li> <li>• <u>External Event mode</u> <ul style="list-style-type: none"> <li>– GPIO_MODE_EVT_RISING : Rising edge trigger detection</li> <li>– GPIO_MODE_EVT_FALLING : Falling edge trigger detection</li> <li>– GPIO_MODE_EVT_RISING_FALLING : Rising/Falling edge trigger detection</li> </ul> </li> </ul>
Pull	Specifies the Pull-up or Pull-down activation for the selected pins. Possible values are: GPIO_NOPULL GPIO_PULLUP GPIO_PULLDOWN
Speed	Specifies the speed for the selected pins Possible values are: GPIO_SPEED_LOW GPIO_SPEED_MEDIUM GPIO_SPEED_FAST

Рисунок 2.17 – Зображення частини списку команд бібліотеки HAL

Бібліотека CMSIS (Cortex Microcontroller Software Interface Standard) більш продуктивна, точна та швидка, використовується для написання процесів, де важливі інтервали часу: генерація ШИМ-сигналів, трактувань та апаратних пауз. Працює з регістрами.

```

66 typedef enum
67 {
68 /***** Cortex-M4 Processor Exceptions Numbers *****/
69 NonMaskableInt_IRQn = -14, /*!< 2 Non Maskable Interrupt */
70 MemoryManagement_IRQn = -12, /*!< 4 Cortex-M4 Memory Management Interrupt */
71 BusFault_IRQn = -11, /*!< 5 Cortex-M4 Bus Fault Interrupt */
72 UsageFault_IRQn = -10, /*!< 6 Cortex-M4 Usage Fault Interrupt */
73 SVCall_IRQn = -5, /*!< 11 Cortex-M4 SV Call Interrupt */
74 DebugMonitor_IRQn = -4, /*!< 12 Cortex-M4 Debug Monitor Interrupt */
75 PendSV_IRQn = -2, /*!< 14 Cortex-M4 Pend SV Interrupt */
76 SysTick_IRQn = -1, /*!< 15 Cortex-M4 System Tick Interrupt */
77 /***** STM32 specific Interrupt Numbers *****/
78 WWDG_IRQn = 0, /*!< Window WatchDog Interrupt */
79 PVD_IRQn = 1, /*!< PVD through EXTI Line detection Interrupt */
80 TAMP_STAMP_IRQn = 2, /*!< Tamper and TimeStamp interrupts through the EXTI line */
81 RTC_WKUP_IRQn = 3, /*!< RTC Wakeup interrupt through the EXTI line */
82 FLASH_IRQn = 4, /*!< FLASH global Interrupt */
83 RCC_IRQn = 5, /*!< RCC global Interrupt */
84 EXTI0_IRQn = 6, /*!< EXTI Line0 Interrupt */
85 EXTI1_IRQn = 7, /*!< EXTI Line1 Interrupt */
86 EXTI2_IRQn = 8, /*!< EXTI Line2 Interrupt */
87 EXTI3_IRQn = 9, /*!< EXTI Line3 Interrupt */
88 EXTI4_IRQn = 10, /*!< EXTI Line4 Interrupt */
89 DMA1_Stream0_IRQn = 11, /*!< DMA1 Stream 0 global Interrupt */
90 DMA1_Stream1_IRQn = 12, /*!< DMA1 Stream 1 global Interrupt */
91 DMA1_Stream2_IRQn = 13, /*!< DMA1 Stream 2 global Interrupt */
92 DMA1_Stream3_IRQn = 14, /*!< DMA1 Stream 3 global Interrupt */
93 DMA1_Stream4_IRQn = 15, /*!< DMA1 Stream 4 global Interrupt */
94 DMA1_Stream5_IRQn = 16, /*!< DMA1 Stream 5 global Interrupt */

```

Рисунок 2.18 – Зображення частини списку команд бібліотеки CMSIS

Основним нововведенням було створення цілого програмного комплексу у купі із візуальним додатком конфігурації налаштування Cube MX. Завдяки ньому вже не потрібно було конфігурувати основні параметри мікроконтролеру самостійно, що зробити на архітектурі від ARM було доволі тяжко. Ускладнювали це завдання велика кількість налаштувань апаратних таймерів, протоколів, портів вводу та виводу, які треба було зібрати у логічний конструкт для забезпечення комфортного програмування.

Таблиця 2.9 – Порівняння програми Cube IDE

Переваги	Недоліки
Вбудований редактор коду, що полегшує роботу з вихідним текстом	Інколи недопрацьований продукт у порівнянні з KEIL uVision через те, що є відносно новим
Відладчик-симулятор, що допомагає налагоджувати роботу програми, що була скомпільована	Менша кількість інформації щодо продукту у порівнянні із KEIL uVision
Вдосконалений менеджер проєктів	-



Підтримка великої кількості операційних систем із якою працює середовище	-
Безкоштовний програмний продукт як для комерційного так і для приватного використання	-
Один і той же виробник програмного продукту що й мікроконтролерів STM	-
Новіший продукт у порівнянні з KEIL uVision	-
Єдине та об'єднане середовище розробки з Cube MX	-

Огляд інтерфейсу та базових функцій інтегрованого середовища розробки Cube IDE:

- 1) Поле написання основної програми. Тут створюється та записується основний її код мовою програмування C.



```

1
2
3
4 * @file      : main.c
5 * @brief    : Main program body
6
7 * @attention
8
9 * <h2><center>&copy; Copyright (c) 2022 STMicroelectronics.
10 * All rights reserved.</center></h2>
11 *
12 * This software component is licensed by ST under BSD 3-Clause license,
13 * the "License"; You may not use this file except in compliance with the
14 * License. You may obtain a copy of the License at:
15 *   opensource.org/licenses/BSD-3-Clause
16 *
17
18 */
19 /* USER CODE END Header */
20 /* Includes -----*/
21 #include "main.h"
22
23 /* Private includes -----*/
24 /* USER CODE BEGIN Includes */
25 #include "stdio.h"
26 #include <string.h>
27 #include "socket.h"
28 #include "w5500.h"
29 #include "rc522.h"
30 #include "rc5221.h"
31 /* USER CODE END Includes */
32
33 /* Private typedef -----*/
34 /* USER CODE BEGIN PTD */

```

Рисунок 2.19 – Зображення робочого поля написання програмного коду в програмі «Cube IDE»

## 2) Швидка панель доступу до основних функцій.



Рисунок 2.20 – Зображення поля швидкого доступу в програмі «Cube IDE»

## 3) Панель доступу до файлів проекту, драйверів бібліотек, конфігураційних файлів та інших.

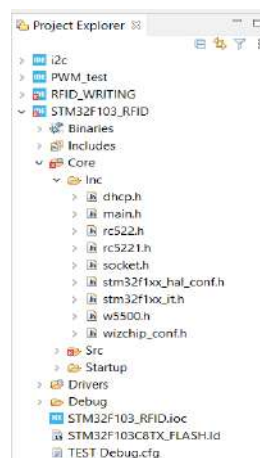


Рисунок 2.21 – Зображення поля доступу файлів в програмі «Cube IDE»

## 4) Вікно консолі налаштування програмного коду.

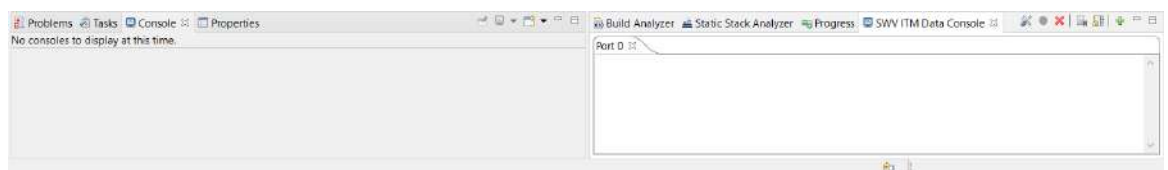


Рисунок 2.22 – Зображення консольного поля для налаштування програмного коду в програмі «Cube IDE»

Cube MX є візуальним графічним редактором для конфігурування мікроконтролерів сімейства STM32, що дозволяє генерувати код на основі мови C, використовуючи для цього графічних помічник. Графічний генератор коду здатен конфігурувати:

- 1) Конфігурація тактування головного обчислювального ядра, тактування дільників та перед дільників, тактування шин, аналогово-цифрових перетворювачів.
- 2) Конфігурація протоколів: SPI, UART, USB, I2C, CAN.
- 3) Конфігурація прямого доступу до пам'яті DMA.
- 4) Конфігурація таймеру підрахунку контрольної суми CRC.
- 5) Конфігурація входів та виходів мікроконтролера.

Огляд інтерфейсу та базових функцій графічного конфігуратора налаштувань Cube MX :

- 1) Розділ вибору існуючих одно кристальних EOM від STM, отримання повної документації на мікроконтролери, огляд вартості в офіційному магазині.



Рисунок 2.23 – Зображення поля вибору мікроконтролера в програмі «Cube MX»

- 2) Головна візуальна панель задання конфігурації пінів входів та виходів мікроконтролеру, що дозволяє бачити фізичну модель корпусів мікроконтролерів та порівнювати із проектом, що знаходиться у розробці.



налаштувати частоту та періоди тактування усіх можливих інформаційних шин мікроконтролера.

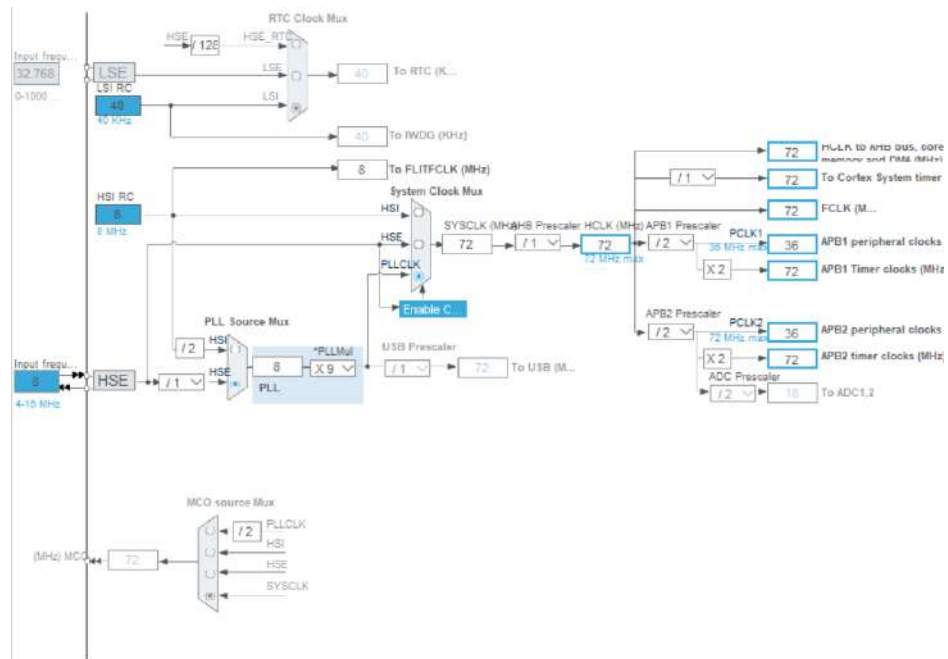


Рисунок 2.26 – Зображення поля налаштування тактування мікроконтролера в програмі «Cube MX»

- 5) Панель налаштувань «менеджер проекту». Завдяки панелі можна визначити ім'я проекту, його розташування, інтегроване середовище розробки, до якого буде генеруватись конфігураційний код та велику кількість доповнюючих налаштувань.

Pinout & Configuration	Clock Configuration
<b>Project</b> Project Settings Project Name: STM32F103_RPID Project Location: C:\Users\User\Desktop\STM32\Proj_1.txt Application Structure: Advanced <input type="checkbox"/> Do not generate the main()	
<b>Code Generator</b> Toolchain Folder Location: C:\Users\User\Desktop\STM32\Proj_1\STM32F103_RPID Toolchain / IDE: STM32CubeIDE <input checked="" type="checkbox"/> Generate Under Root	
<b>Advanced Settings</b> Linker Settings Minimum Heap Size: 0x200 Minimum Stack Size: 0x400 Thread-safe Settings Cortex-M3NS <input type="checkbox"/> Enable multi-threaded support Thread-safe Locking Strategy: Default – Mapping suitable strategy depending on RTOS selection.	
<b>MCU and Firmware Package</b> MCU Reference: STM32F103C8Tx Firmware Package Name and Version: STM32Cube_FW_F1_V1.8.4 <input checked="" type="checkbox"/> Use latest available version <input checked="" type="checkbox"/> Use Default Firmware Location: C:\Users\User\STM32Cube\Repository\STM32Cube_FW_F1_V1.8.4	

Рисунок 2.27 – Зображення поля конфігурації проекту в програмі «Cube MX»

б) Панель математичного розрахунку та налаштувань живлення мікроконтролера та окремих його елементів, враховуючи тактову частоту усіх елементів та температурні режими роботи процесора. Будівання діаграм.

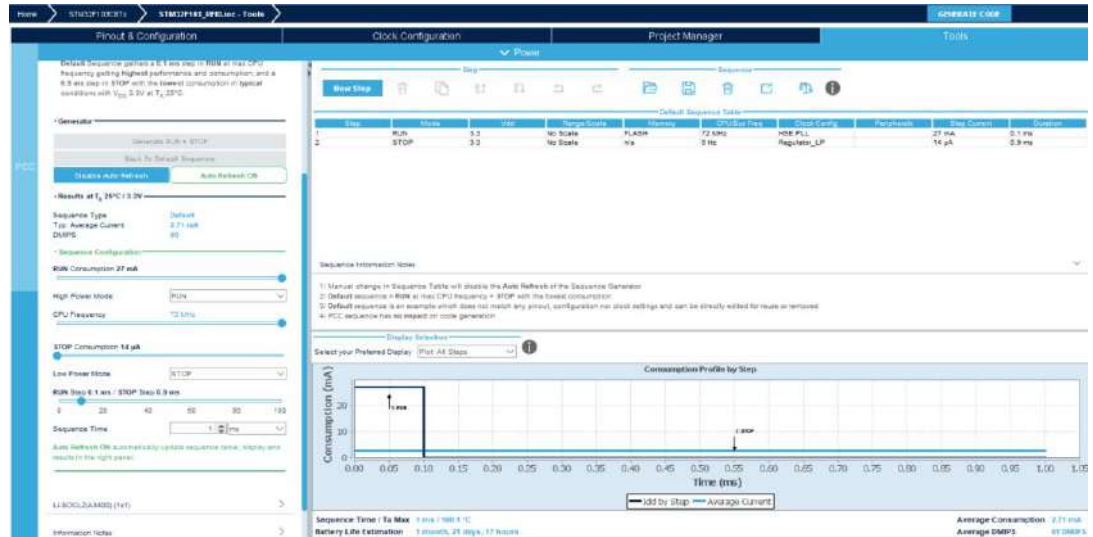


Рисунок 2.28 – Зображення поля конфігурації мікроконтролера в програмі «Cube MX»

Таблиця 2.10 – Порівняння програми Cube MX

	Cube MX
Переваги	Недоліки
Наявність бази даних усіх мікроконтролерів від виробника STM	Сумісність тільки з інтегрованим середовищем розробки Cube IDE
Наявність всієї технічної документації до кожного мікроконтролера	
Вбудований інтерфейс конфігурації тактування шин, таймерів та ядра для подальшої генерації програмного коду	

Вбудований інтерфейс конфігурації портів вводу та виводу для подальшої генерації програмного коду	
Вбудований інтерфейс конфігурації протоколів зовнішнього зв'язку для подальшої генерації програмного коду	
Зручний візуальний спосіб налаштувань усієї периферії контролера	

Проаналізувавши можливі популярних середовищ розробки для створення програмної частини для програмування виконавчого пристрою системи контролю та управління доступом було вирішено використовувати інтегроване середовище розробки від виробника мікроконтролеру STM Cube IDE, та задля спрощення і запобігання помилок під час процесу налаштування, графічний генератор коду Cube MX.

### 2.2.2 Програмне забезпечення для програмування блоку серверної частини

Будова серверної частини- це найбільш накопичена щодо програмного коду частина прототипу автоматизованої системи контролю та управління доступом. Саме у цьому структурному блоці зберігається та обробляється уся корисна інформація, що потрібна для автоматизації процесу контролю доступу, а саме: ідентифікаційні номери користувачів та відвідувачів, статус, у якому вони знаходяться, дати подій та час. Саме програма серверної частини дозволяє виконавчому пристрою відкрити доступ.

Для коректної роботи серверної частини було обрано використовувати саме такі сучасні програмні рішення, а саме: середу розробки «Microsoft Visual Studio 2022», веб-інтерфейс адміністрування систем управління базами даних

«PHPMyAdmin», збірку веб-серверу «XAMPP», мову високо-рівнявого програмування «C#», мову програмування «SQL», реляційну систему управління базами даних «MySQL».

Microsoft Visual Studio 2022- це сучасна лінійка продуктів компанії «Microsoft», що єднає у собі інтегровану середу розробки та інші продукти.

За допомогою інтегрованого середовища розробки можливо зручно писати та змінювати програмний код на мові програмування «C#».

Під час проходження практики використовувалася нова версія програми «Microsoft Visual Studio 2022» розширення «Community», що є безоплатною версією, завантаженою на офіційному сайті проекту.

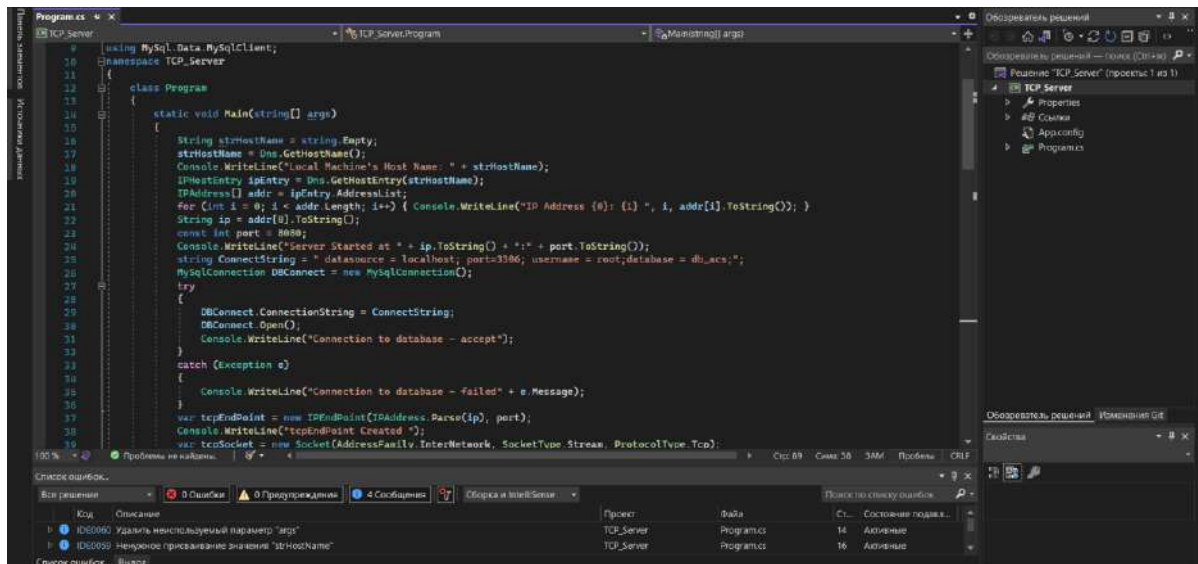


Рисунок 2.29 – Зображення робочого поля програмування в програмі «Microsoft Visual Studio 2022»

Веб-інтерфейс адміністрування систем управління базами даних «PHPMyAdmin»- це веб-додаток з відкритим кодом, написаний мовою PHP- є веб-інтерфейсом для адміністрування систем управління базами даних (СУБД) «MySQL». «PhpMyAdmin» дозволяє за допомогою браузера здійснювати адміністрування сервера «MySQL», запускати команди мовою «SQL», переглядати вміст таблиць та баз даних.



Під час проходження переддипломної практики, веб-додаток «PhpMyAdmin» є одним із найбільш популярних інструментів для керування таблицями баз даних.

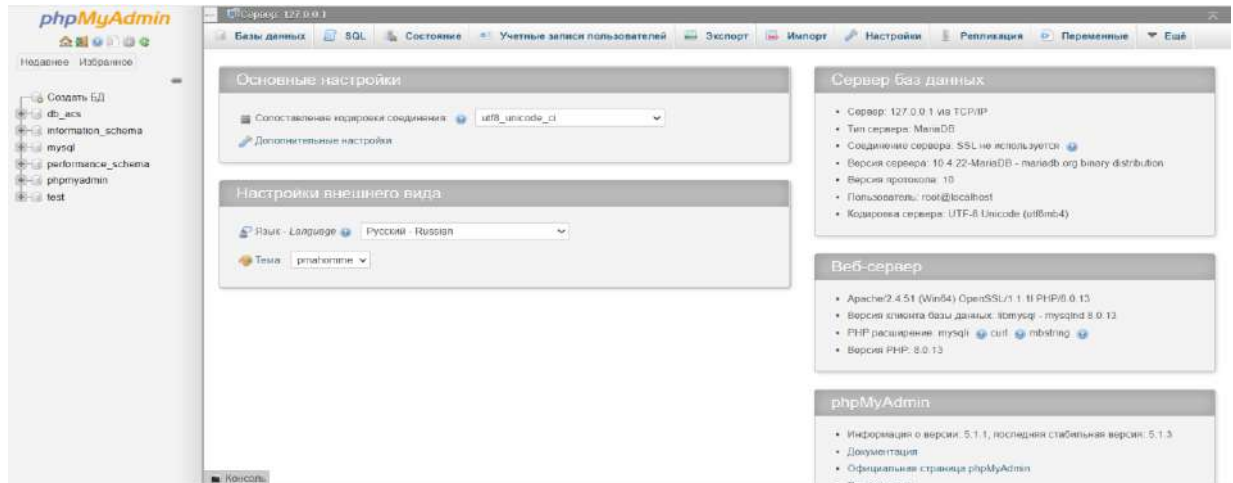


Рисунок 2.30 – Зображення робочого поля програми «phpMyAdmin»

Збірка веб-серверу «ХАМРР»- це між-платформна збірка веб-серверу, що поєднує в собі «Apache», «MySQL», інтерпретатор скриптів «PHP», що дозволяють створювати повноцінний локальний сервер. Є однією з найбільш популярних щодо використання збірок.

«ХАМРР»- працює з усіма 32-розрядними ОС Microsoft (98[3]/2000/XP/2003/Vista/7), а також з Linux, Mac OS X і Solaris. Програма вільно розповсюджується згідно з ліцензією GNU General Public License і є безкоштовним, зручним у роботі web-сервером, здатним обслуговувати динамічні сторінки.

На сьогоднішній день «ХАМРР» є однією з найкращих збірок веб-сервера, за допомогою цієї збірки можна швидко розгорнути на своєму комп'ютері повноцінний та швидкий веб-сервер.

Повний пакет збірки «ХАМРР» включає в себе:

- 1) Web-сервер Apache з підтримкою SSL
- 2) СУБД MySQL
- 3) PHP

- 4) Perl
- 5) FTP-сервер FileZilla
- 6) POP3/SMTP сервер
- 7) Утиліту phpMyAdmin

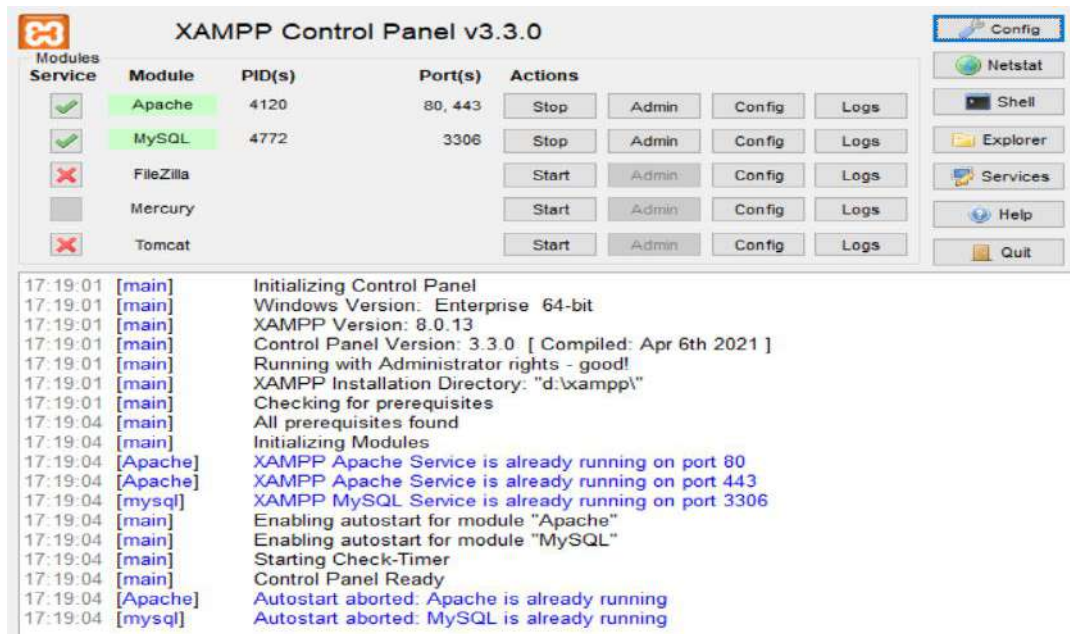


Рисунок 2.31 – Зображення робочого поля конфігурації вбудованих додатків в програмі «XAMPP»

«C#» (або «C Sharp»)- це об'єктно-орієнтована мова програмування високого рівня, що була створена інженерами компанії «Microsoft». За синтаксисом мова подібна до мов програмування «C» та «C++», а саме така мова («C») використовується під час програмування мікроконтролера STM32, котрий є частиною апаратної реалізації прототипу проекту автоматизованої системи контролю та управління доступом.

Через велику кількість синтаксичних конструкцій, ця мова є потужним інструментом та заощаджує сили під час конструювання алгоритмів, які на інших мовах займатимуть набагато більше часу.

Мова програмування «C#» є одна з найбільш надійних методів програмування через те, що задіє віртуальний процесор під час запуску написаних програм, тож при виникненні помилок вони не поширюються на інші програми, що працюють паралельно.

Також, мова програмування «C#» має потужний набір бібліотек, які використовуються під час роботи із серверними завданнями.

Під час проходження практики було звернено до українського інтернет видання «dou.ua», проведення незалежного опитування продемонструвало, що у 2022 році програмування на «C#» залишається популярним та займає друге місце у загальному рейтингу.

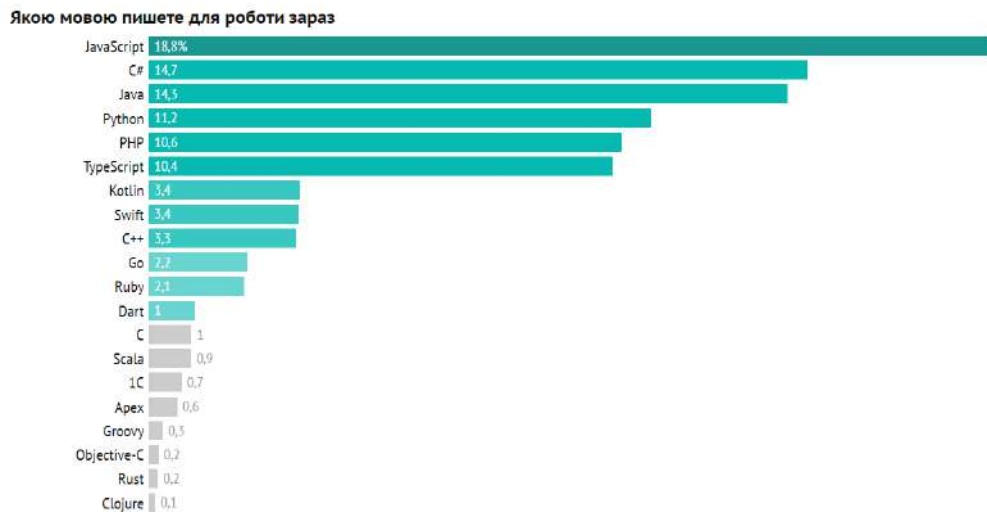


Рисунок 2.32 – Зображення статистики використання мов програмування серед розробників

«SQL»- декларативна мова програмування, що створена для роботи із реляційними базами даних. На відміну від «C#», є інформаційно-логічною мовою програмування, що створена для опису, вилучення та внесення корективів у дані, що знаходяться у реляційній базі даних.

За допомогою неї у базі даних можна зручно створювати таблиці, змінювати та вносити в них дані, змінювати їх структуру.

Згодом SQL покращився, збагатився новими конструкціями, забезпечив можливість опису та управління новими об'єктами, що зберігаються (наприклад, індекси, уявлення, тригери і процедури, що зберігаються) - і став набувати рис, властивих мовам програмування.

При всіх своїх змінах SQL залишається найпоширенішим лінгвістичним засобом взаємодії прикладного програмного забезпечення з базами даних. У

той самий час сучасні СУБД, і навіть інформаційні системи, використовують СУБД, надають користувачеві розвинені засоби візуального побудови запитів.

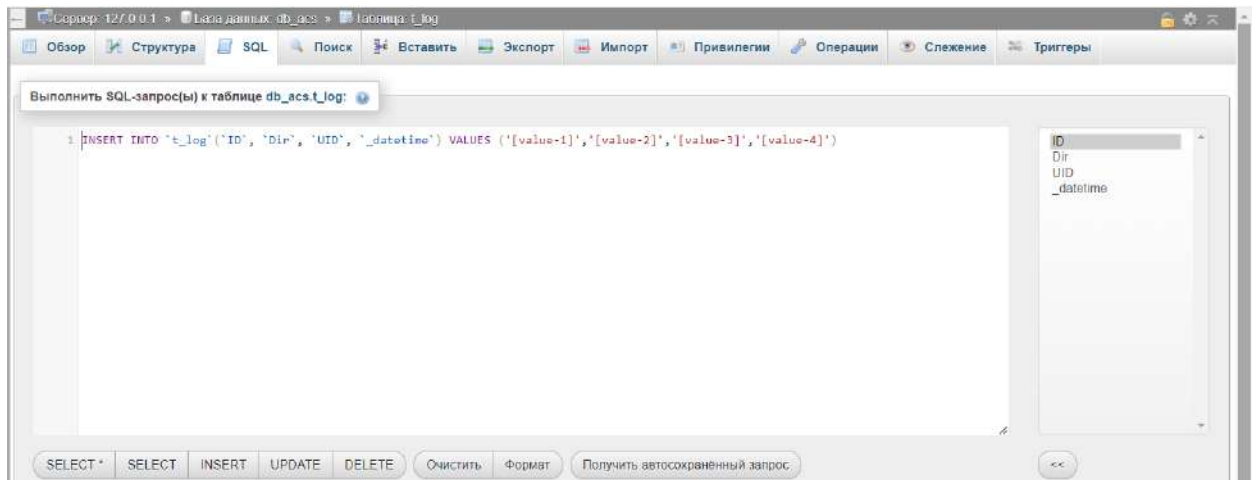


Рисунок 2.33 – Зображення робочого поля програмування в програмі «phpMyAdmin»

«MySQL»- це вільна реляційна система управління базами даних. Інструмент йде у пакеті програмного серверного комплексу «ХАМРР». Є рішенням для малих та середніх додатків, що якнайкраще підходить по концепції взаємодії «клієнт-частини» автоматизованої системи контролю та управління доступом із «сервером».

Розробку та підтримку MySQL здійснює корпорація Oracle, яка отримала права на торгову марку разом із поглиненою Sun Microsystems, яка раніше придбала шведську компанію MySQL AB. Продукт розповсюджується як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність на замовлення ліцензійних користувачів. Саме завдяки такому замовленню майже в ранніх версіях з'явився механізм реплікації.



Рисунок 2.34 – Зображення емблеми додатку «MySQL» від компанії «Oracle»

### 2.3 Висновки за розділом

Під час виконання другого розділу було визначено та обґрунтовано перелік електронних приладів та компонентів, що мають бути використані в проекті СКУД складських приміщень. Також, був розглянутий список програмного забезпечення та додатків, які сприятимуть процесу розробки програмних кодів серверної та виконавчої частин та їх налаштувань.

## РОЗДІЛ 3

### ПРОГРАМНО-АПАРАТНА РОЗРОБКА ТА ВИПРОБУВАННЯ

#### 3.1 Проектування СКУД

Проектування системи контролю та управління доступом складських приміщень- це предметна область, що потребує найбільшої уваги та терпіння. Після щільного аналізу програмних, апаратних комплексів, вибору об'єктів для дослідження та розробки- можна починати складання системи.

Завданням до розробки СКУД є:

- 1) Створення програмно/апаратного блоку виконавчого пристрою або «клієнта»
- 2) Створення програмно/ апаратного блоку пристрою внесення та корегування інформації ідентифікаційних номерів
- 3) Створення програмного блоку серверної частини або «сервера»
- 4) Створення комп'ютерного додатку до пристрою внесення та корегування інформації ідентифікаційних номерів

#### 3.2 Проектування виконавчого пристрою

Проектування та побудова апаратної частини СКУД ґрунтується на потребах системи в цілому. На тому де і як вона буде працювати, які режими роботи зустрічатимуться, які впливи навколишнього світу загрожуватимуть. Цьому процесу треба приділити увагу, бо електроніка має багато «підводного каміння», яке обов'язково треба врахувати та уникнути під час розробки та проектування.

Основу електронної компонентної бази виконавчого пристрою, з якої складається система, було розкрито у минулому розділі, а саме: мікроконтролер STM32F103, мікросхему підтримки мережі WIZnet W5500, Мікросхему безконтактного зчитування електронних даних MFRC522 та чіп

«смарт-карт» від Mifare Classic. Також, було розглянуто технологію збереження даних електронного ключа.

Мета: розглянути важливі електронні компоненти, які сприятимуть правильній роботі електронної схеми виконавчого приладу СКУД, створити електронну схему виконавчого приладу СКУД, створити проект друкованої плати з розгляданням процесу конфігурації налаштувань для цього.

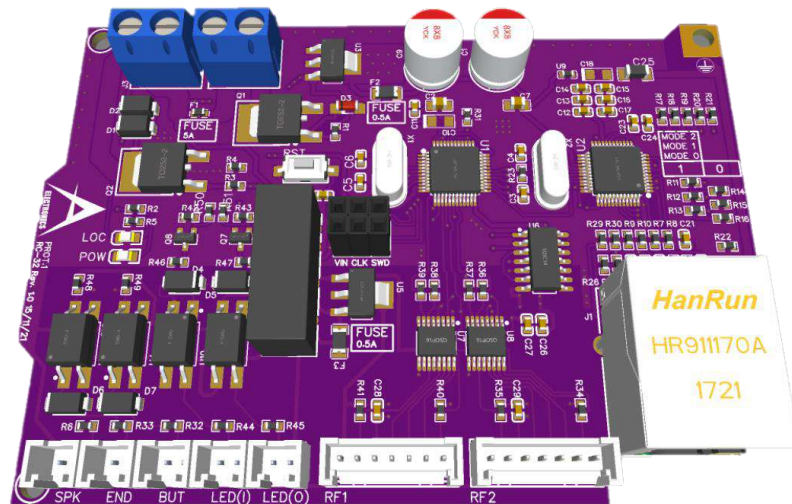


Рисунок 3.1 – Зображення моделі друкованої плати контролера СКУД

Перш за все, треба проаналізувати умови використання системи, яка є системою захисту та контролю. Перед тим, як вона почне процес контролю, її треба захистити саму. Насамперед, як і будь-який електронний прилад, що має велику кількість електронних напівпровідників та комунікує із іншим приладом методом фізичного поєднання їх струмопровідних провідників, прилад «боїться» впливу електричного струму на електромагнітних перешкод.

Через це, обов'язковими є вузли захисту від зовнішніх впливів, оптичні, що мають діелектричний проміжок.

Перш за все, вузли з'єднання електронних приладів безконтактного зчитування MFRC 522 із головним мікроконтролером STM32F103, для чого були обрані мікросхеми MAX 14850, які здатні «розв'язати», за допомогою оптичного бар'єру, інформаційні шини мікроконтролера із зовнішнім світом.

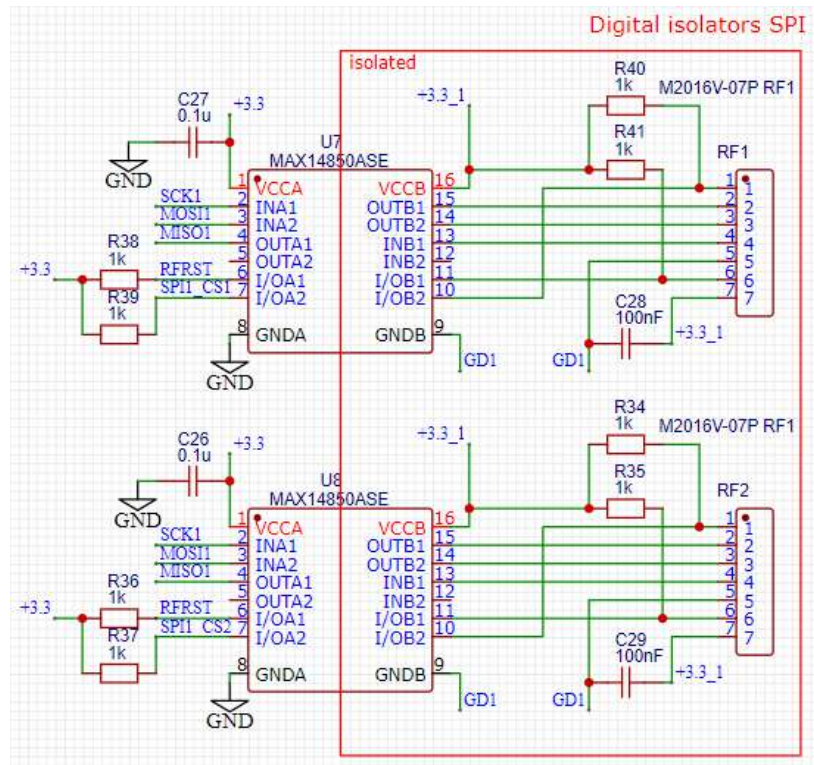


Рисунок 3.2 – Зображення частини діелектричного захисту принципової електронної схеми

По-друге, виконавчий пристрій має органи керування: кнопки, індикацію та кінцевий вимикач. Вони також безпосередньо контактують із периферійними контактами мікроконтролера, де будь-який несанкційних вплив на них може вивести мікросхему із робочого стану.

Щоб уникнути цього- використовуються оптичні пари PC 817. Вони також мають бар'єр від струмопровідних частин на зовні.

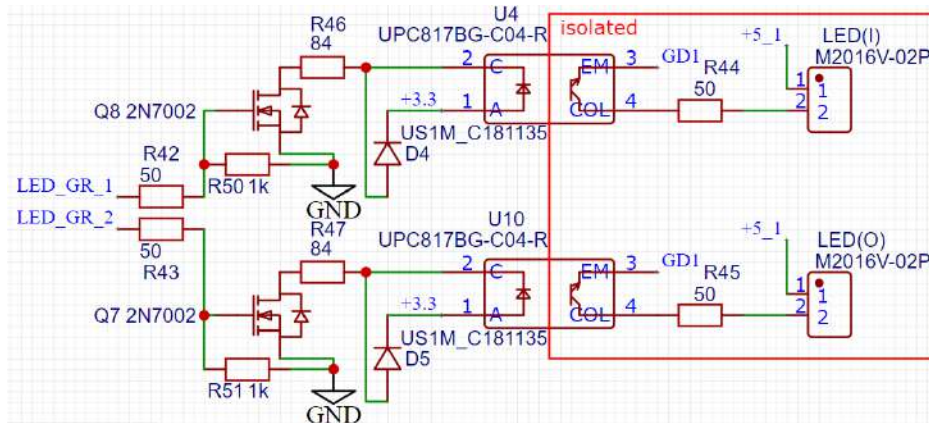


Рисунок 3.3 – Зображення частини діелектричного захисту принципової електронної схеми



Буферна мікросхема 74HC125D. Запобігає несвоєчасному спрацюванню електронних приладів безконтактного зчитування на мікросхемі MFRC 522. Виключає можливість виникнення помилок на інформаційній шині SPI. За допомогою неї відбувається «спілкування» зчитувачів із основним мікроконтролером.

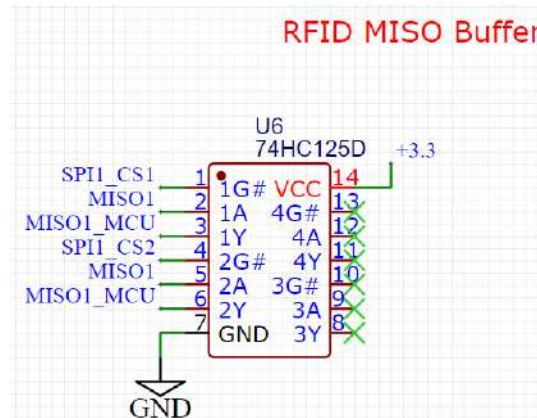


Рисунок 3.4 – Зображення частини буферної мікросхеми принципової електронної схеми

Живиться схема від напруги 12 вольт, через те, що більшість пристроїв фізичного запобігання проникненню працюють від цієї напруга, вона у більшості випадків є універсальною. Сама ж логічна частина схеми, починаючи від мікроконтролера, закінчуючи контролером локальної мережі живиться від напруги логічного рівня 3,3 вольт. Для цього потрібен вузол стабілізації напруги. Його параметри мають відповідати нормам низької пульсації постійної напруги для запобігання потрапляння перешкод на лінію живлення, що допоможе уникнути можливих проблем із працездатністю обчислювальних пристроїв та передавачів.

Вузол перетворення та стабілізації напруги побудований на мікросхемі AMS 1117. Перед вузлом стабілізації, згідно стандартам, присутня схема запобігання так званої «переполіусовки» живлення, коли при підключення плутаються провідники за полюсами. Також, розташовано силовий запобіжник для захисту сильно-струмової частини схеми та логічний

запобіжник, що має захистити схему обчислювальних блоків у випадку виникнення нестандартних ситуацій.

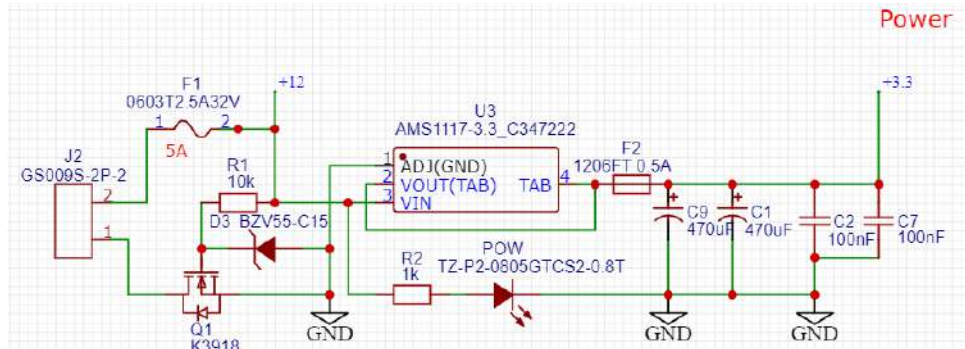


Рисунок 3.5 – Зображення частини блоку живлення контролера СКУД  
принципової електронної схеми

Звісно ж, захист живлення присутній і на приладах безконтактного зчитування, що розташовані не на друкованій платі виконавчого пристрою, а зовні. В якості ізолятора живлення- ізоляційний перетворювач імпульсно-трансформаторного типу з гальванічною «розв'язкою» виходу В1205, стабілізатор AMS 1117.

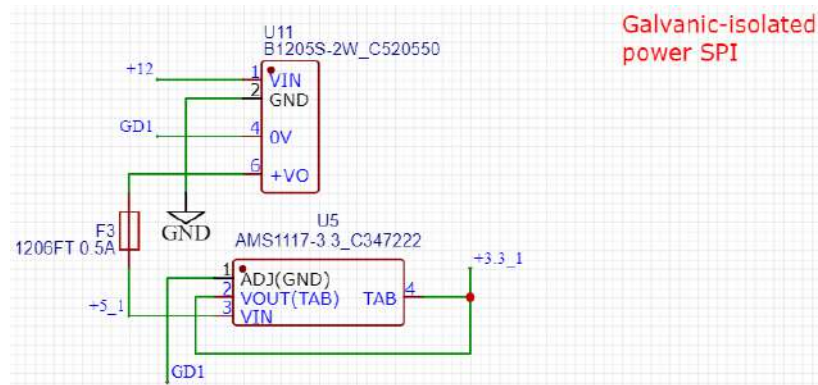


Рисунок 3.6 – Зображення частини блоку живлення периферії принципової  
електронної схеми

Вузол комутації зовнішнього навантаження, а саме електричного магніту, під який проектувався пристрій заснований на силовому транзисторі MOSFET-типу K3918, що здатен до комутації великого струму.

Перспективним етапом подальшого розвитку є додавання контрольно-вимірювальної функції наявності навантаження. Це допомогло б системі контролювати увесь час присутність навантаження на виході приладу СКУД.

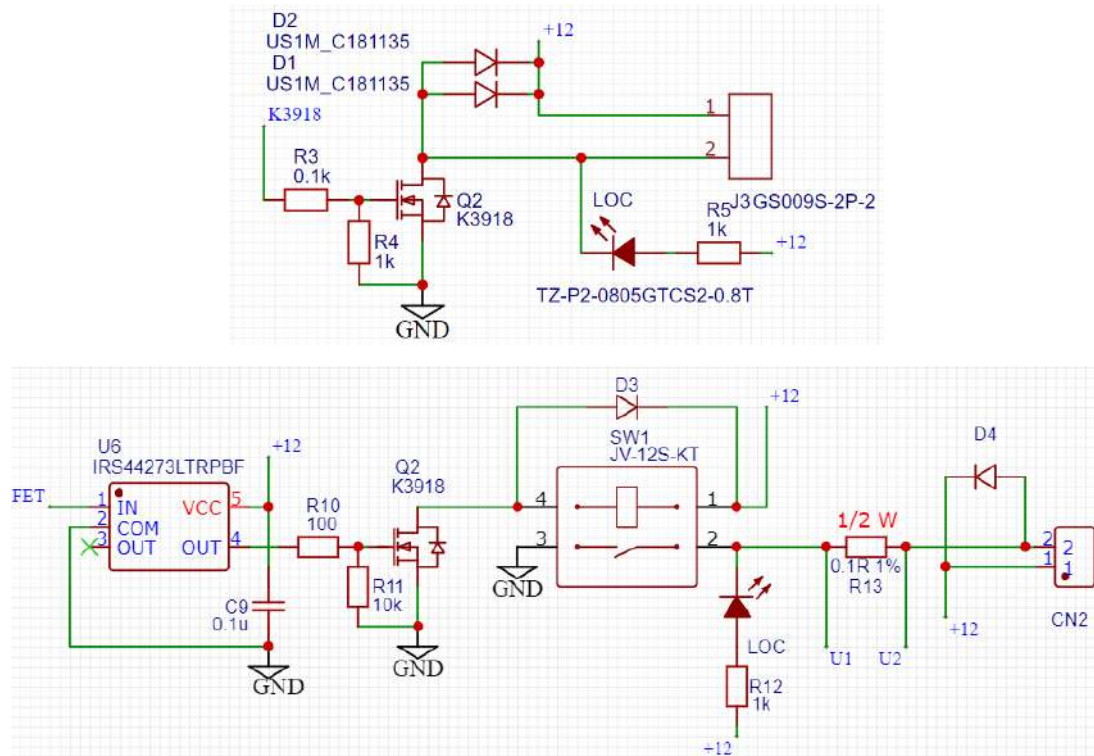


Рисунок 3.7 – Зображення частини комутаційного вузла принципової електронної схеми

WIZnet W5500- це вбудований Ethernet-контролер з підключенням TCP/IP, який забезпечує спрощене підключення до інтернету або локальної мережі для вбудованих систем за допомогою SPI (Serial Peripheral Interface).

Особлива увага приділена до якості вузла живлення контролера. Мається роз'єднувальний трансформатор, що ізолює з'єднання Ethernet фізичного протоколу.



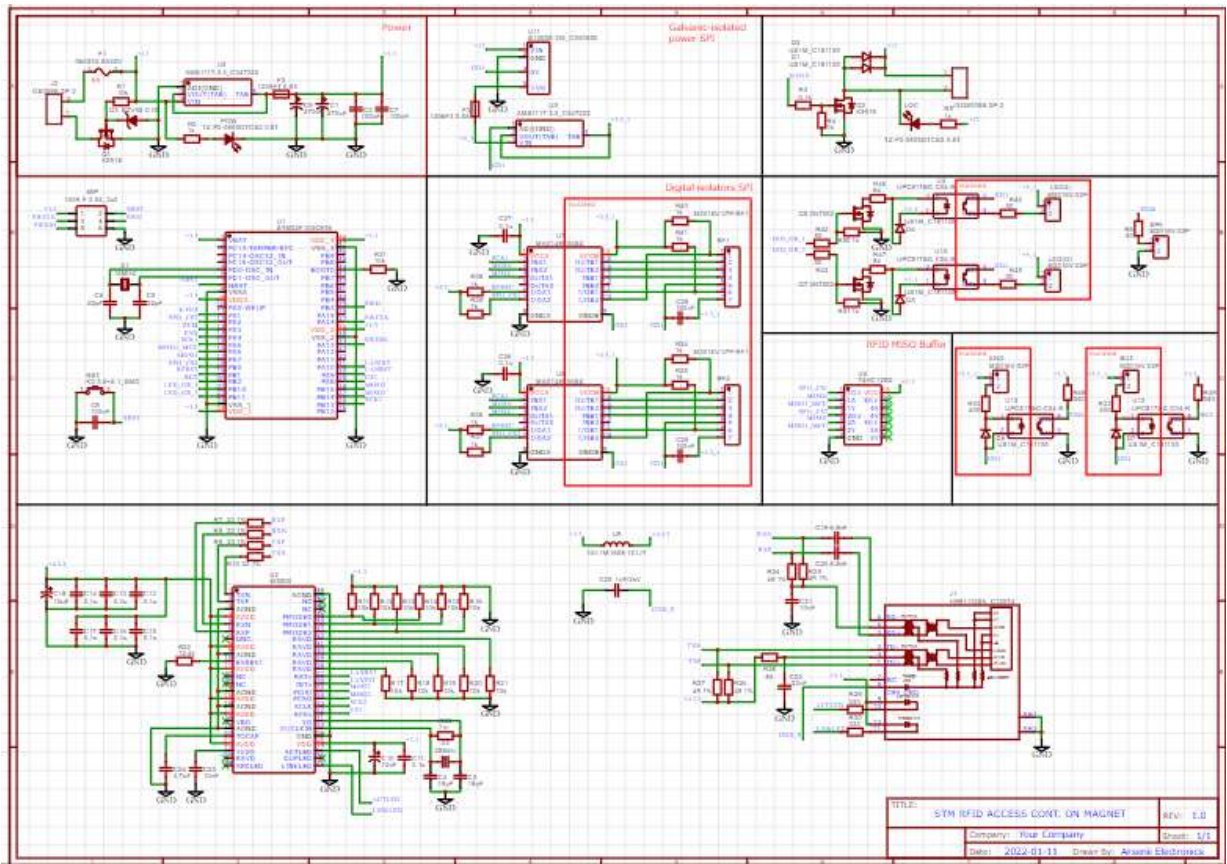


Рисунок 3.10 – Зображення принципової електронної схеми СКУД

EasyEda- крос-платформне веб-орієнтоване середовище автоматизації проектування електроніки, що включає редактор принципів схем, редактор топології друкованих плат, SPICE-симулятор, хмарне сховище даних, систему управління проектами, а також засоби підготовки та замовлення виготовлення друкованих плат.

Після створення електричної схеми СКУД розробляється друкована плата, що фізично відображає спроектовану електричну схему. Використовуються всі загальні засоби та правила проектування друкованих плат. Наприкінці ДП конвертується в формат Gerber для числового програмного виготовлення на спеціалізованому виробництві.

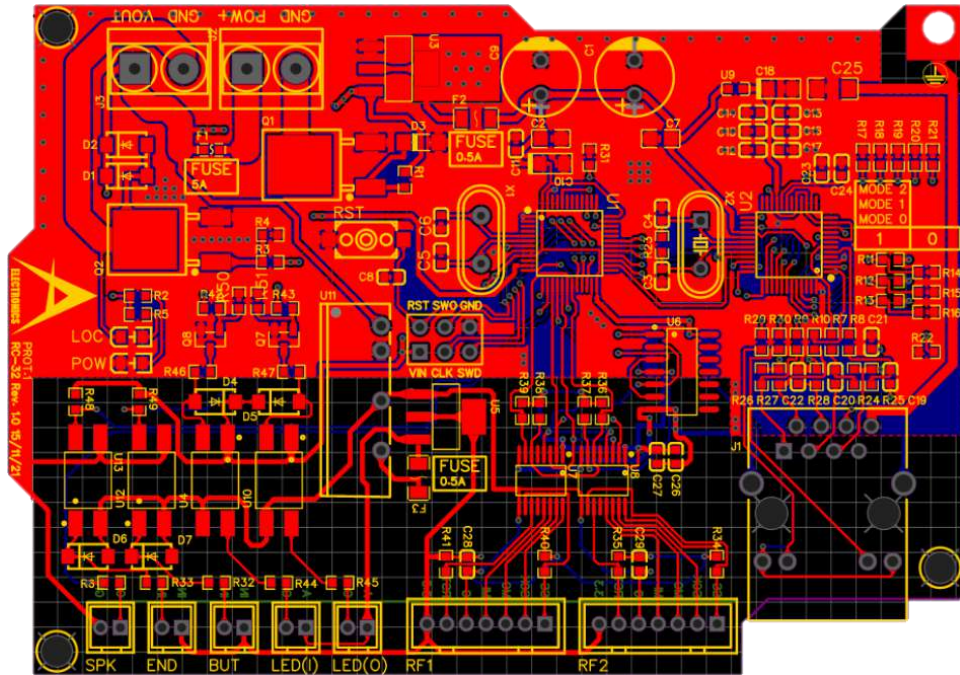


Рисунок 3.11 – Зображення моделі двовимірної моделі друкованої плати в системі автоматизованого проектування

Після виробництва отримуємо готову друковану плату для подальшого монтажу електронних деталей за схемою. Монтаж відбувається спеціалізованим обладнанням за технологією SMD (поверхової) прив'язки до плати.

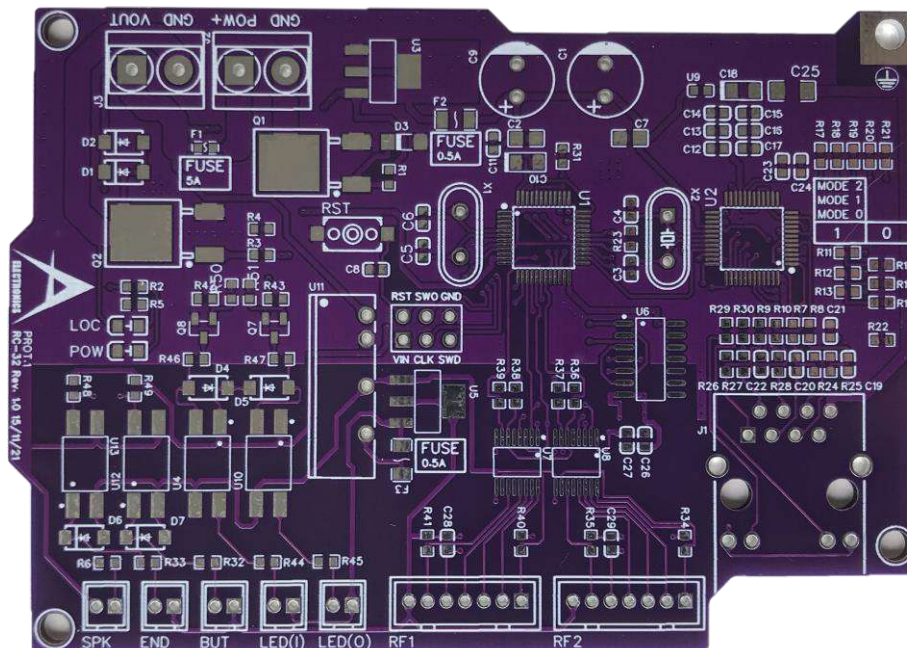


Рисунок 3.12 – Зображення друкованої плати СКУД

### 3.3 Проектування пристрою внесення даних

Проектування та побудова апаратної частини пристрою внесення даних ґрунтується на потребах системи вносити та корегувати ідентифікаційні дані користувачів в систему таблиць бази даних.

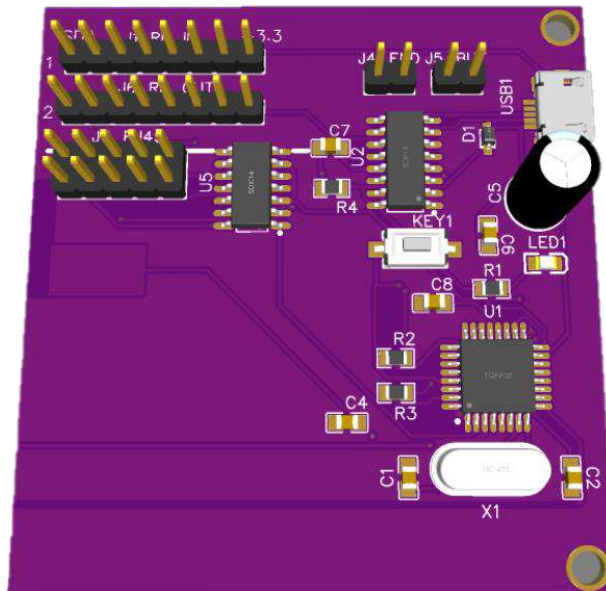


Рисунок 3.13 – Зображення моделі пристрою внесення даних

Основу електронної компонентної бази пристрою внесення даних, з якої складається система, було розкрито у минулому розділі, а саме: мікроконтролер ATmega 328, мікросхема безконтактного зчитування даних MFRC522 та мікросхема-перетворювач інтерфейсів UART-USB CH340.

Перший електронний вузол приладу зчитування та корегування даних-вузол живлення. Великих потреб до нього не представляється, через те, що прилад працює в умовах офісного простору біля основного комп'ютера бази даних, де зберігається уся інформація. Окрім цього- персональний комп'ютер і без того має гарні показники якості струму на виходах інформаційних портів USB, тому можна обійтися елементарною базою.

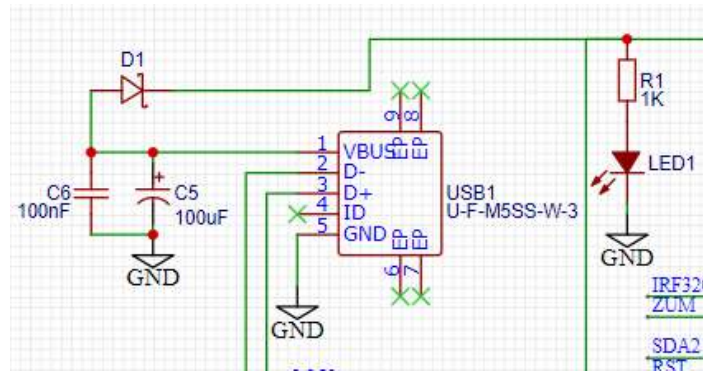


Рисунок 3.14 – Зображення частини блоку живлення принципової електронної схеми

Другий вузол, це вузол обчислення, зчитування та зв'язку.

Мікроконтролер тактується зовнішнім кварцовим резонатором на 16 МГц. Інформаційна шина приладу безконтактного зчитування під'єднана до апаратного протоколу зовнішньої комунікації SPI. Працює на низькій швидкості, тому що високої у даному випадку не потребує.

Мікросхема-перетворювач протоколів USB-UART під'єднана до двох виходів прийому та передачі даних (RX/TX). Також, вона виконує функцію знижувального стабілізатора напруги до 3,3 вольт для живлення модуля безконтактного зчитувача MFRC522 та модуля програмування для мікроконтролера.

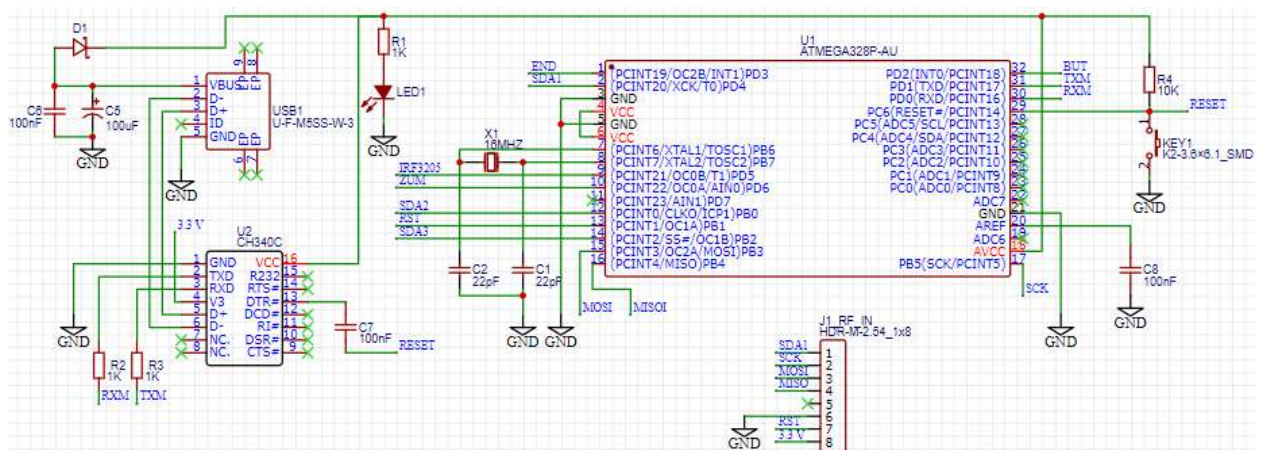


Рисунок 3.15 – Зображення принципової електронної схеми



Як і у випадку виконавчого пристрою «клієнта», проектується друкована плата, спираючись на електричну схему пристрою. Таким чином компонується Gerber-файл проекту та віддається на виробництво.

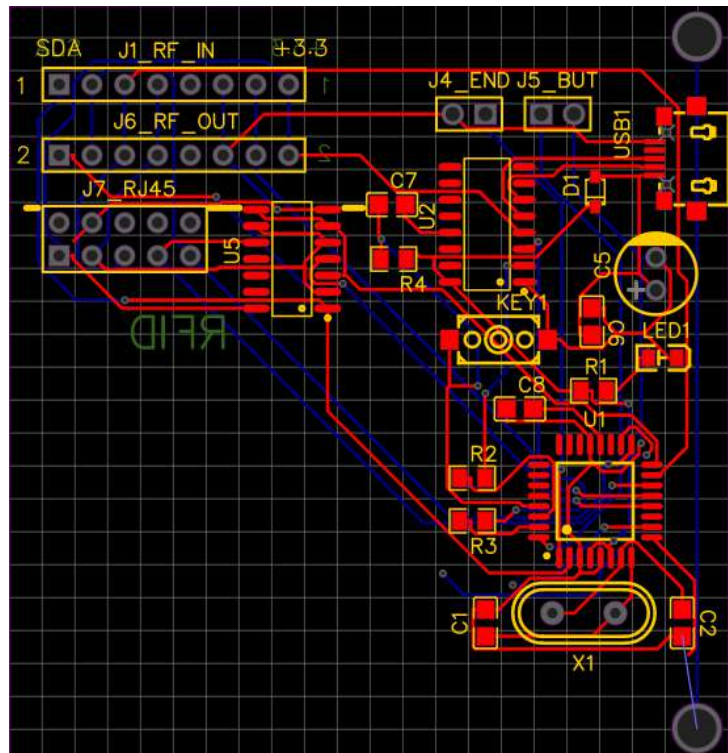


Рисунок 3.16 – Зображення двовимірної моделі друкованої плати в системі автоматизованого проектування

### 3.4 Програмування комплексу виконавчого пристрою

Як було описано раніше, для проектування програмної частини виконавчого пристрою СКУД було обрано використовувати інтегроване середовище розробки Cube IDE та візуальний генератор конфігураційного програмного коду Cube MX.

Спочатку, створюється проект у генераторі конфігураційного програмного коду Cube MX. Обирається визначений мікроконтролер STM32F103. Надається значення усім потрібним контактам логічного входу та виходу, таким як: виходи світлодіодів індикатора, вихід звукового індикатора на основі п'єзо-випромінювача, вихід до силового транзистору К3918, входи кнопки виходу та кінцевого вимикача.

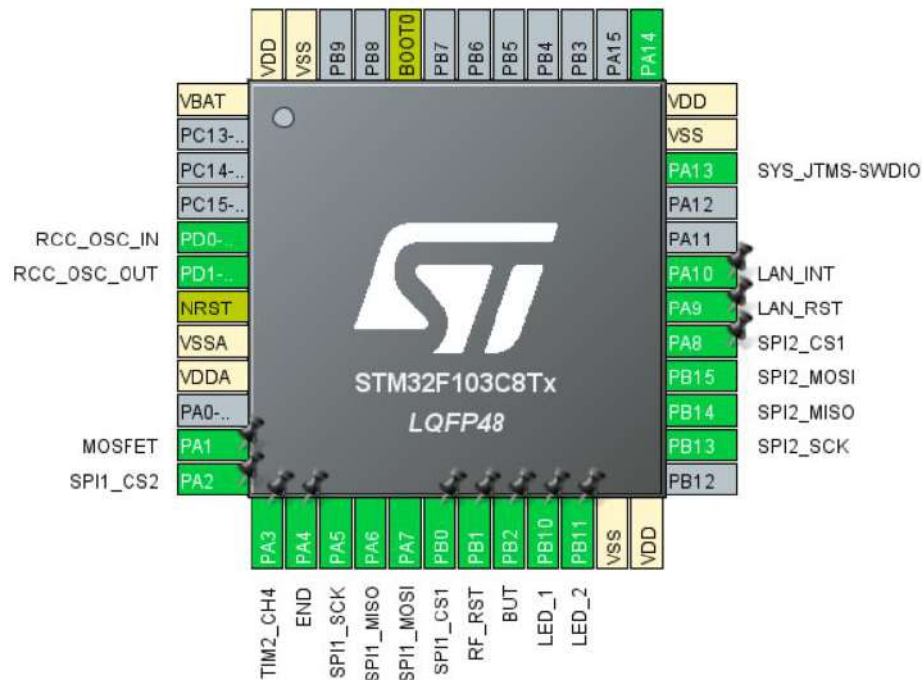


Рисунок 3.17 – Зображення конфігураційної моделі мікроконтролера в програмі «CubeMX»

Визначаються джерело, параметри тактування головного ядра та інформаційних шин. Обирається тактування від кварцового резонатора на 8 МГц. Частота роботи процесора мікроконтролера 72 МГц, частота тактування шин 36 та 72 МГц.

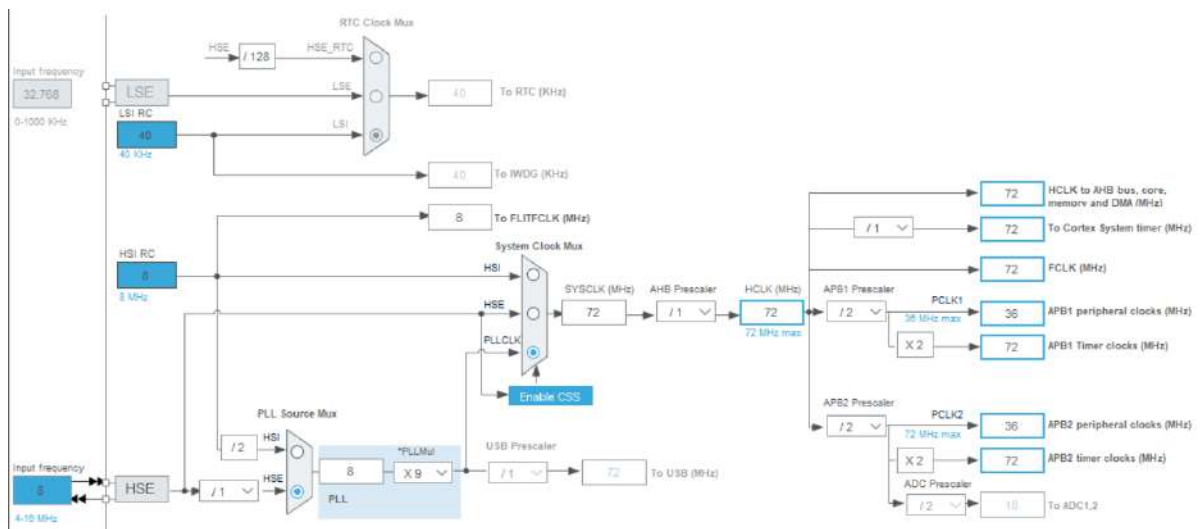


Рисунок 3.18 – Зображення конфігураційної панелі тактування мікроконтролера

Визначаються основні характеристики протоколу SPI до пристроїв зчитування на мікросхемах MFRC 522:

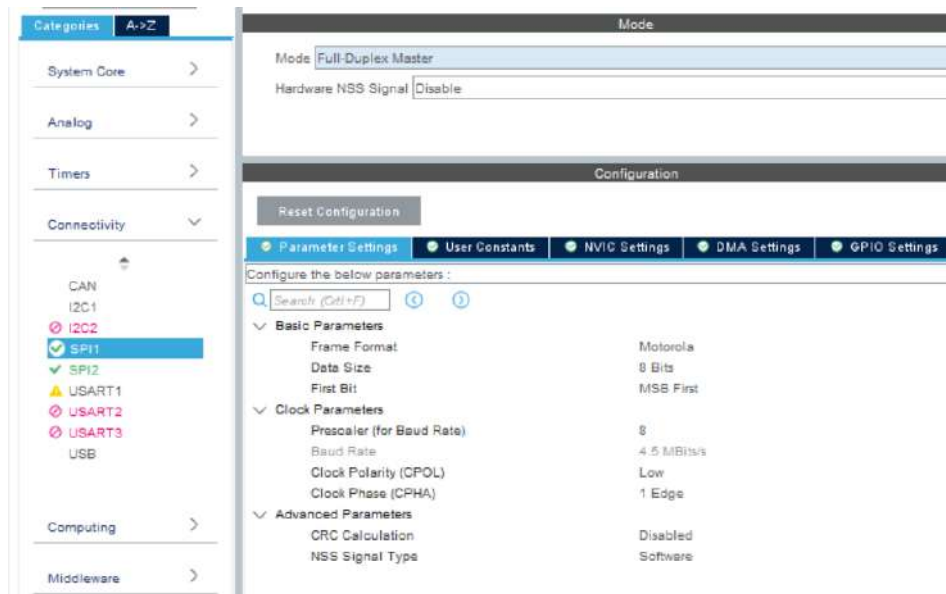


Рисунок 3.19 – Зображення налаштування фізичного протоколу із MFRC522 в програмі «CubeMX»

Таблиця 3.1 – Опис характеристик протоколу SPI1

	<b>SPI 1</b>
Режим роботи	Повно дуплексний (Full Duplex Master)
Розмір пакету даних	8 біт
Перший біт	MSB First
Дільник тактування шини	8
Швидкість тактування шини	4,5 МГц (4,5 Мб/с)
Фаза такту	1 Edge

Визначаються основні характеристики протоколу SPI до пристрою з'єднання мережі Ethernet:

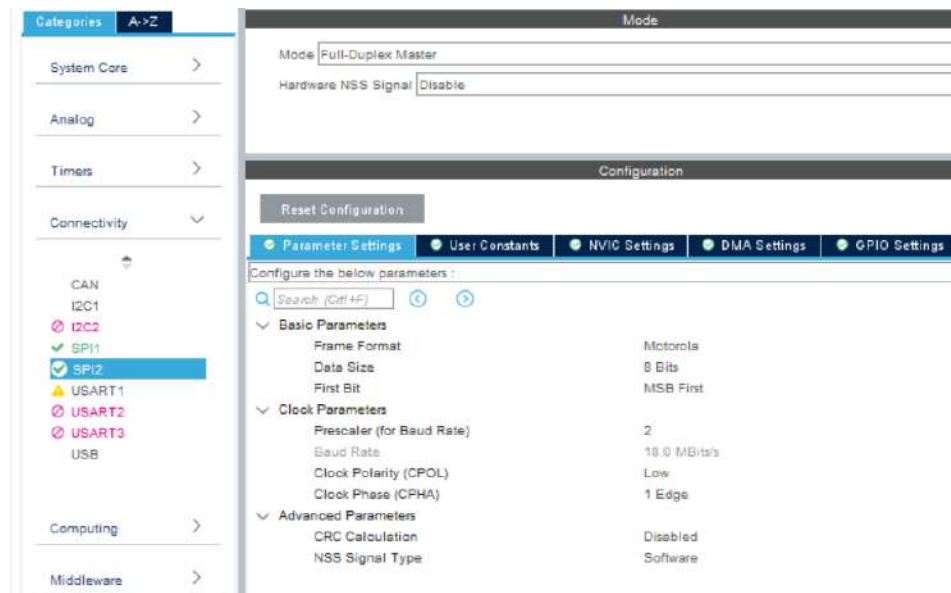


Рисунок 3.20 – Зображення налаштування фізичного протоколу із W5500 в програмі «CubeMX»

Таблиця 3.2 – Опис характеристик протоколу SPI2

	<b>SPI 2</b>
Режим роботи	Повно дуплексний (Full Duplex Master)
Розмір пакету даних	8 біт
Перший біт	MSB First
Дільник тактування шини	2
Швидкість тактування шини	18 МГц (18 Мб/с)
Фаза такту	1 Edge

Визначаються характеристики конфігурації внутрішнього базового таймеру для роботи з часовими інтервалами та широтно-імпульсна модуляція для роботи з п'єзо-випромінювачем.

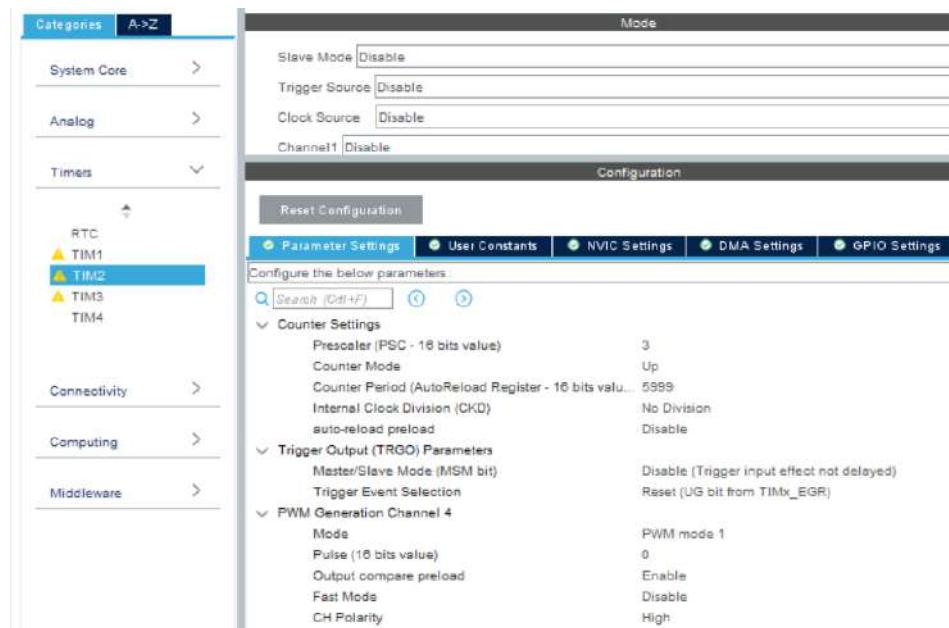


Рисунок 3.21 – Зображення налаштування базового таймеру в програмі «CubeMX»

Таблиця 3.3 – Опис характеристик базового таймера

	<b>Base Timer</b>
Режим роботи	Лічильник в більшу сторону
Дільник шини тактування	3
Період рахунку	6000
Режим ШІМ (модуляції)	PWM Mode 1
Канал ШІМ	4 канал

Налаштування проекту, а саме надання йому ім'я, визначення директорії знаходження, вибір інтегрованого середовища розробки для генерації програмного коду та інші.

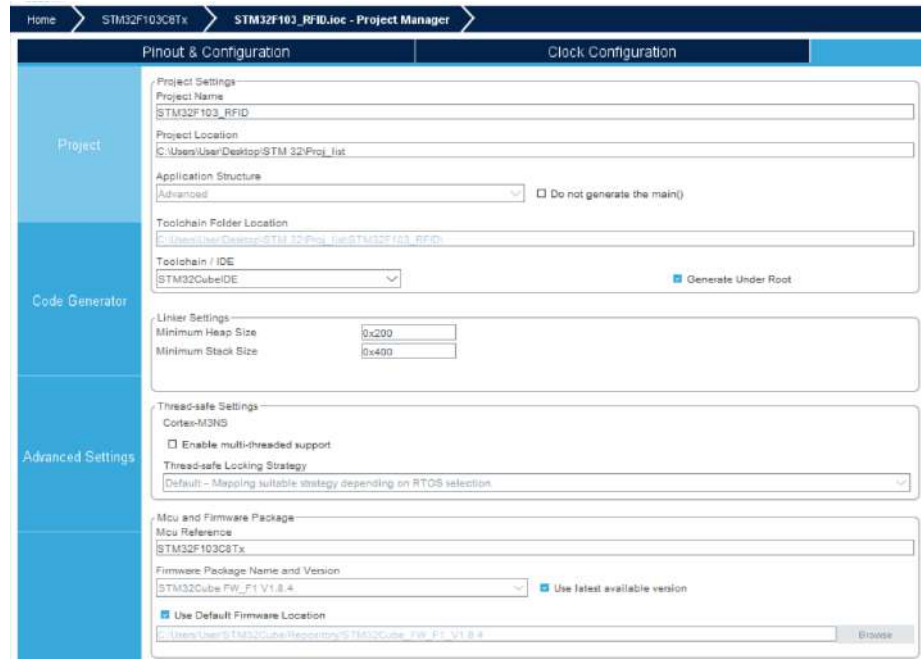


Рисунок 3.22 – Зображення налаштування проекту в програмі «CubeMX»

Далі натисканням кнопки «Generate Code» генерується стартові конфігураційні налаштування апаратного рівня до середовища «Cube IDE» мовою C, потім відкриває його.

Для коректного програмування пристрою треба визначити бібліотеки пристроїв, які мають протоколи з'єднання та вбудований контролер, що використовує ці протоколи. Бібліотеки: RC 522, Socket, W 5500, Wizchip\_conf. Файли для будовання бібліотек надаються виробниками мікросхем MFRC522 та W5500.

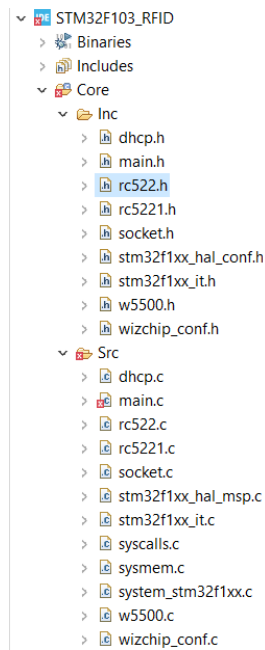


Рисунок 3.23 – Зображення налаштування проекту в програмі «CubeMX»

Програмування, як було вказано раніше, відбувається мовою С. Мова С-компільована статично типізована мова програмування загального призначення, розроблена в 1969-1973 роках співробітником Bell Labs Денісом Рітчі як розвиток мови Бі.

Систематично використовується програмними інженерами вбудованих систем для програмування одно кристальних ЕОМ завдяки гарним характеристикам компіляторів та зручному й потужному синтаксису написання програмного коду.

По-перше, визначаються бібліотеки, що додалися у файл проекту:

Лістинг 3.1 – Додавання бібліотек на початку проекту

```
#include "main.h"
#include "stdio.h"
#include <string.h>
#include "socket.h"
#include "w5500.h"
#include "rc522.h"
#include "rc5221.h"
```

«main.h» додає заголовний файл основного коду програми. Він зберігає головні ідентифікатори портів вводу та виводу мікроконтролера. Це полегшує

роботу із портами під час програмування завдяки інтуїтивно зрозумілому формуванню.

Бібліотека "stdio.h"- це заголовний файл стандартної бібліотеки мови C, за допомогою якої можна працювати із вводом та виводом даних в консоль та буфери. Має вбудовані макроси, структури та типи даних для формування правильного запиту для стандартних операцій.

Бібліотека <string.h>- також заголовний файл стандартної бібліотеки мови C, за допомогою якої є можливість працювати із вводом та виводом рядків, що закінчуються та 0.

Бібліотека "socket"- надається виробником контролеру Ethernet від WizNET. Враховує всі базові функції для можливості роботи із програмними сокетамі в системі, такі як: підключення, прийом та передача даних.

Бібліотека "w5500"- також надається виробником контролеру Ethernet та знаходиться ц відкритому до завантаження доступі. Має функції для роботи із апаратними налаштуваннями контролеру зв'язку, є основною бібліотекою для з'єднання частини клієнта та сервера.

"rc522" та "rc5221"- бібліотеки для роботи із мікросхемою радіочастотної ідентифікації MFRC522. Створено 2 різних класи для спрощення роботи із кожним ідентифікатором окремо. Класи мають одні й ті ж функції для роботи з чіпом. Створені за допомогою технічної документації мікросхем із описом базових функцій роботи з ними.

Додаються ідентифікатори для спрощення інтуїтивного розуміння коду та можливості зручного коректування інформації в процесі налагодження приладу:

### Лістинг 3.2 – Визначення назв констант

```
#define DATA_BUF_SIZE  2048
#define TCP_SOCKET      0
#define PORT_TCPS       5000
#define DATA_BUF_SIZE  2048
uint8_t gDATABUF[DATA_BUF_SIZE];
```



Створюється масив даних налаштувань із мережевими конфігураціями (фізичка адреса Ethernet- контролеру, IP- адреса, маска підмережі, адреса маршрутизатору мережі) та визначаються базові апаратні функції для роботи із мережевим пристроєм WizNET за протоколом SPI.

Фізична адреса Ethernet-контролеру надається власноруч, варіативність за бажанням, а не як зазвичай, прописується виробником мікросхеми під час виробництва.

### Лістинг 3.3 – Конфігурування змінних

```
wiz_NetInfo gWIZNETINFO = {.mac = {0x80, 0xfa, 0x5b, 0x6a, 0xe2, 0xd1},
    .ip = {192, 168, 31, 103},
    .sn = {255, 255, 255, 0},
    .gw = {192, 168, 31, 1},
    .dns = {0, 0, 0, 0},
    .dhcp = NETINFO_DHCP};
void W5500_Select(void){
    HAL_GPIO_WritePin(GPIOA, SPI2_CS1_Pin, GPIO_PIN_RESET); //correct}
void W5500_Unselect(void){
    HAL_GPIO_WritePin(GPIOA, SPI2_CS1_Pin, GPIO_PIN_SET); //correct}
void W5500_ReadBuff(uint8_t* buff, uint16_t len){
    HAL_SPI_Receive(&hspi2, buff, len, HAL_MAX_DELAY);}
void W5500_WriteBuff(uint8_t* buff, uint16_t len){
    HAL_SPI_Transmit(&hspi2, buff, len, HAL_MAX_DELAY);}
uint8_t W5500_ReadByte(void){
    uint8_t byte;
    W5500_ReadBuff(&byte, sizeof(byte));
    return byte;}
void W5500_WriteByte(uint8_t byte){
    W5500_WriteBuff(&byte, sizeof(byte));}
uint8_t stat;
```

Визначаються приватні функції, масиви, цілочисельні на символічні типи даних до зберігання ідентифікаційних номерів, конфігурації мережевого пристрою та інші. Також, функції ініціалізації портів вводу-виводу та протоколів зв'язку та передачі даних, що визначаються автоматично конфігуратором CubeMX.

### Лістинг 3.4 – Ініціалізація структурних блоків, визначення змінних

```
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_SPI2_Init(void);
```

```

static void MX_TIM2_Init(void);
uint8_t blockAddr;
uint8_t RC_size;
uint8_t status;
uint8_t      str[MFRC522_MAX_LEN];
uint8_t sn[5];
uint8_t sn1[5];
uint8_t bo = 0;
char  snbuff[64];
char  snbuff1[64];
char  statbuff[64];
char  statbuff1[64];
uint8_t i;

```

Визначаються основні функції обробники з'єднання із програмним сокетом, отримання та відправки через сокет для першого «клієнт» (вхідного) та другого «клієнт» (вихідного) пристроїв, функції обробки вхідної інформації.

Сокет (Socket)- це назва програмного інтерфейсу забезпечення обміну даними між процесами. Процеси при такому обміні можуть виконуватися як на одному пристрої обміну, так і різних, пов'язаних між собою мережею. Сокет - абстрактний об'єкт, що представляє кінцеву точку з'єднання.

Для взаємодії між машинами за допомогою стека протоколів TCP/IP використовуються адреси та порти. Адреса є 32-бітною структурою для протоколу IPv4.

Номер порту – ціле число в діапазоні від 0 до 65535 (для протоколу TCP).

Лістинг 3.5 – Створення функції передачі даних за допомогою TCP-сокету

```

void SendSocket(void){
stat = socket(TCP_SOCKET, Sn_MR_TCP, 8080, 0);
uint8_t addr[4] = {192,168,31,203};
stat = connect(TCP_SOCKET, (uint8_t*)addr, 8080);
HAL_Delay(5);
send(0, (uint8_t*)statbuff, strlen(statbuff));
send(0, (uint8_t*)snbuff, strlen(snbuff));
HAL_Delay (5);
uint8_t rc[32];
recv(0, ((uint8_t*)rc), 32);
char  rcbuff[10];
sprintf(rcbuff, "%c", rc[0]);
if (rc[0] == 'R') {
HAL_GPIO_WritePin(GPIOA, MOSFET_Pin, GPIO_PIN_SET);
for(i=0; i<=2; i++){
HAL_TIM_PWM_Start (&htim2, TIM_CHANNEL_4);

```

```

HAL_Delay(50);
HAL_TIM_PWM_Stop (&htim2, TIM_CHANNEL_4);
HAL_Delay(100);}
HAL_Delay(1550);
HAL_GPIO_WritePin(GPIOA, MOSFET_Pin, GPIO_PIN_RESET);
}else{
HAL_Delay(2000);}
disconnect(TCP_SOCKET);
close(TCP_SOCKET);
HAL_Delay(5);}

```

У циклі коду «main» визначаємо функції, що виконуються один раз при старті системи.

По-перше, відбувається ініціалізація портів входу та виходу, ініціалізація протоколів передачі даних SPI, ініціалізація всіх периферійних пристроїв та базових системних таймерів, лічильників та інших.

По-друге, після запуску протоколів, передаються дані до мікросхеми мережевого зв'язку, визначаються буферні простори для зберігання інформації, що передається від «клієнта» до «сервера».

Запускається стартовий звуковий сигнал на виході звукового індикатора.

### Лістинг 3.6 – Стартовий блок програмного коду

```

MX_GPIO_Init();
MX_SPI1_Init();
MX_SPI2_Init();
MX_TIM2_Init();
HAL_Delay(200);
for (i=0; i<=2; i++){
HAL_TIM_PWM_Start (&htim2, TIM_CHANNEL_4);
HAL_Delay(50);
HAL_TIM_PWM_Stop (&htim2, TIM_CHANNEL_4);
HAL_Delay(100);}
HAL_Delay(500);
HAL_TIM_PWM_Start (&htim2, TIM_CHANNEL_4);
HAL_Delay(500);
HAL_TIM_PWM_Stop (&htim2, TIM_CHANNEL_4);
HAL_GPIO_WritePin(GPIOB, RF_RST_Pin, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, LAN_RST_Pin, GPIO_PIN_RESET);
HAL_Delay(1);
HAL_GPIO_WritePin(GPIOA, LAN_RST_Pin, GPIO_PIN_SET);
HAL_Delay(1000);
reg_wizchip_cs_cbfunc(W5500_Select, W5500_Unselect);
reg_wizchip_spi_cbfunc(W5500_ReadByte, W5500_WriteByte);
reg_wizchip_spiburst_cbfunc(W5500_ReadBuff, W5500_WriteBuff);

```

```
uint8_t rx_tx_buff_sizes[] = {2,2,2,2,2,2,2,2};
wizchip_init(rx_tx_buff_sizes, rx_tx_buff_sizes);
wizchip_setnetinfo(&gWIZNETINFO);
ctlnetwork(CN_SET_NETINFO, (void*) &gWIZNETINFO);
HAL_Delay(1000);
```

Наприкінці- основний цикл програмного коду «While». У ньому процеси відбуваються за без кінцевим циклом. Сюди завантажуються функції перевірки програмних сокетів та їх відправка, аналізується інформація, що надходить з приладів безконтактного зчитування на мікросхемах MFRC522. Також, вписуються основні апаратні затримки без кінцевого циклу.

Процеси утворюються та обробляються постійно, поки ввімкнено живлення основного мікроконтролера STM32F103.

### Лістинг 3.7 – Основний блок циклічного програмного коду

```
while (1){
MFRC522_Init();
MFRC5221_Init();
status = MI_ERR;
status = MFRC522_Request(PICC_REQIDL, str);
status = MFRC522_Anticoll(sn);
if (status == MI_OK) {
sprintf(snbuff, "%x%x%x%x\r\n", sn[0],sn[1],sn[2],sn[3]);
sprintf(statbuff, "ENTER\r\n");
SendSocket() ;
HAL_Delay(5);}
status = MI_ERR;
status = MFRC522_Request1(PICC_REQIDL, str);
status = MFRC522_Anticoll1(sn1);
if (status == MI_OK) {
sprintf(snbuff1, "%x%x%x%x\r\n", sn1[0],sn1[1],sn1[2],sn1[3]);
sprintf(statbuff1, "EXIT\r\n");
SendSocket1() ;
HAL_Delay(5);}
```

### 3.5 Програмування обробника бази даних

З'єднання серверу та клієнту, а саме виконавчого блоку на мікроконтролері, відбувається за протоколом з'єднання TCP (Transmission Control Protocol). Він є одним з головних протоколів передачі даних у мережі інтернет. До його переваги належить велика надійність до передачі та прийому «пакетів» даних. Призначений для керування передачею даних інтернету.

Пакети TCP називаються сегментами. У стеку протоколів TCP/IP виконує функції транспортного рівня моделі OSI.

Основний алгоритм роботи за допомогою протоколу TCP у Ethernet з'єднанні полягає у тому, що «клієнт» (одиниця зв'язку) відкриває запит до «серверу» (інша одиниця зв'язку), яка в свою чергу прослуховує запити до неї. Після з'єднання між двома учасниками зв'язку проходить передача потрібних даних, а саме, відправлення на серверну частину зчитаного ідентифікаційного коду виконавчим пристроєм. Після цього сервер закриває встановлений зв'язок на руйнує Socket з'єднання. TCP здійснює надійну передачу потоку байтів від одного процесу до іншого.

Перший етап, це встановлення інтегрованого середовища розробки Visual Studio 2022, додатків та їх налаштування. Програмування відбуватиметься за допомогою високорівневої мови програмування C#.

Створюється консольний додаток із назвою TCP\_Server, що виконуватиме обов'язки програми-конектора із базою даних MySQL. Після цього додається список директорій, які забезпечують можливість використання готових структур та функцій для подальшого написання програмного коду за допомогою їх можливостей.

### Лістинг 3.8 – Додавання бібліотек на початку проекту

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using MySql.Data.MySqlClient;
```

Клас System означає, що використовується бібліотека System у проекті. Це дає деякі корисні класи та функції, такі як клас Console або функція/метод WriteLine.

Клас System.Data дозволяє об'єкту реалізувати DataAdapter і представляє набір методів і властивостей, пов'язаних із дією, які

використовуються для заповнення та оновлення набору даних та оновлення джерела даних.

Клас `System.Collections.Generic` забезпечує загальну реалізацію стандартної структури даних, як-от пов'язані списки, стеки, черги та словники.

Клас `System.Linq` надає різні типи класів і методів, які підтримують запити LINQ.

Клас `System.Text` містить класи, які представляють кодування символів ASCII та Unicode. Абстрактні базові класи для перетворення блоків символів у блоки байтів та з них і допоміжний клас, який маніпулює та форматує об'єкти `String` без створення проміжних екземплярів `String`.

Клас `System.Threading.Tasks` надає типи, які спрощують роботу з написанням паралельного та асинхронного коду.

Клас `System.Net` означає, що використовується бібліотека для розробки програм Windows Store або настільних програм.

Клас `System.Net.Sockets`- це підключення бібліотеки для роботи із програмними сокетом. Програмування сокетів - це спосіб з'єднання двох вузлів мережі для зв'язку один з одним. Це одностороннє налаштування клієнта та сервера, де клієнт підключається, надсилає повідомлення на сервер, а сервер показує їх за допомогою підключення через сокет.

Далі потрібно встановлення бібліотеки `MySQL.Data.MySqlClient`, за допомогою якої програма Dot Net матиме змогу зв'язку із сервером MySQL. Після встановлення бібліотеки з офіційного сайту MySQL, вона так само додається до проекту за допомогою функції `using`.

Розглянемо основні функції алгоритму взаємодії процесів в програмному кодї, що відповідають за процес підключення, відправки та отримання даних за допомогою сокет-з'єднання.

По-перше, перед переходом до основного циклу, в циклі `main`, отримаємо ім'я хосту (ім'я машини), якою розгорнуто серверний блок та виведемо його у строку за допомогою консольного додатку.

## Лістинг 3.9 – Блок отримання ім'я хосту

```
String strHostName = string.Empty;
strHostName = Dns.GetHostName();
Console.WriteLine("Local Machine's Host Name: " + strHostName);
```

По-друге, отримаємо адресу IP машини двох основних видів: версій ipv4 та ipv6 та виведемо їх у строку за допомогою консольного додатку. Якщо кількість адресів більша за 2- запускається функція вибору потрібної адреси. У даному випадку система працює з використанням адреси версії ipv4, її технічних можливостей для роботи в локальній мережі достатньо.

## Лістинг 3.10 – Блок отримання IP- адреси

```
for (int i = 0; i < addr.Length; i++) { Console.WriteLine("IP Address {0}:
{1} ", i, addr[i].ToString()); }
String ip;
if (addr.Length > 2){
Console.Write("There is more than one ipv4 address. Please select number =");
int i = int.Parse(Console.ReadLine());
ip = addr[i].ToString();
}else{
string st1 = addr[0].ToString();
string st2 = addr[1].ToString();
if (st1.Length < st2.Length) ip = addr[0].ToString(); else ip =
addr[1].ToString();}
```

Далі вмикається процес авторизації, де необхідно ввести логін та пароль, щоб перейти до блоку підключення з базою даних MySQL. Також, формується рядок SQL-запросу, що буде відправлена. Вона містить: порт підключення, ім'я бази даних, до якої є потреба підключення та авторизаційні дані. Після введення даних та проходження авторизації очищаємо консольний вивід.

Відправляємо запит із вказаними даними за вказаними реквізитами. Якщо з'єднання трапилось- до консолі надсилається відповідна рядок узгодження, якщо відбулась помилка- надсилається рядок, що сповіщає про помилку.

## Лістинг 3.11 – Блок авторизації

```
Console.Write("Input login:");
string login = Console.ReadLine();
Console.Write("Input password:");
```

```

string password = Console.ReadLine();
string ConnectString;
if (password == ""){
ConnectString = " datasource = localhost; port=3306; username = " + login +
";database = db_acs;";
}else{
ConnectString = " datasource = localhost; port=3306; username = " + login +
";password="+password+";database = db_acs;";}
Console.Clear();
MySQLConnection DBConnect = new MySqlConnection();

```

В основному програмному циклі while за допомогою локальних змінних створюємо буферний простір для зберігання даних, що будуть надходити від пристрою зчитування. Вводимо програму у режим сокет прослуховування, коли машина очікуватиме надходження запиту за з'єднання від клієнта.

### Лістинг 3.12 – Блок визначення буферів

```

var bufferSTAT = new byte[256];
var bufferUID = new byte[256];
var sizeSTAT = 0;
var sizeUID = 0;

```

Далі відбувається процес прийому надходження даних від пристрою зчитування. Для зберігання даних застосовується вище створені буферні простори bufferSTAT та bufferUID.

### Лістинг 3.13 – Блок прийому даних

```

do{
sizeSTAT = listener.Receive(bufferSTAT);
dataSTAT.Append(Encoding.UTF8.GetString(bufferSTAT, 0, sizeSTAT));
sizeUID = listener.Receive(bufferUID);
dataUID.Append(Encoding.UTF8.GetString(bufferUID, 0, sizeUID));}while
(listener.Available > 0);

```

Після процесу прийому даних відбувається алгоритм порівняння ідентифікаційних даних, що надійшли, із даними, що зберігаються у таблиці бази даних. Надається відповідь до частини клієнту у випадку співпадання даних із наявними у базі даних.



## Лістинг 3.14 – Блок порівняння ідентифікаційних даних

```

MySQLCommand command = new MySQLCommand("SELECT * FROM t_uid WHERE Status=1
AND UID = @a;", DBConnect);
command.Parameters.Add("@a", MySQLDbType.VarChar).Value = UIDstr;
adapter.SelectCommand = command;
adapter.Fill(table);
if (table.Rows.Count > 0){
int employID = int.Parse(table.Rows[0][2].ToString());
Console.WriteLine("Active status of "+table.Rows[0][1].ToString()+" Accept!");
listener.Send(Encoding.ASCII.GetBytes("R"));
listener.Shutdown(SocketShutdown.Both);
listener.Close();
}

```

Цикл алгоритму обробки закінчується відправкою даних до бази даних, що надаються при вході або виході, та запис їх у згідні таблиці, а саме: напрям руху об'єкту, ідентифікаційний номер картки, ідентифікаційний номер терміналу, дата та час події.

## Лістинг 3.15 – Блок відправки даних до БД

```

comman.Parameters.Add("@d", MySQLDbType.VarChar).Value = direction;
comman.Parameters.Add("@t", MySQLDbType.VarChar).Value = termID;
comman.Parameters.Add("@e", MySQLDbType.VarChar).Value = employID;
comman.Parameters.Add("@u", MySQLDbType.VarChar).Value = UIDstr;
adapte.SelectCommand = comman;
adapte.Fill(temp);

```

Також, створено наявне відображення алгоритму роботи серверної частини за допомогою блок-схеми, що описує послідовність роботи програмного коду.

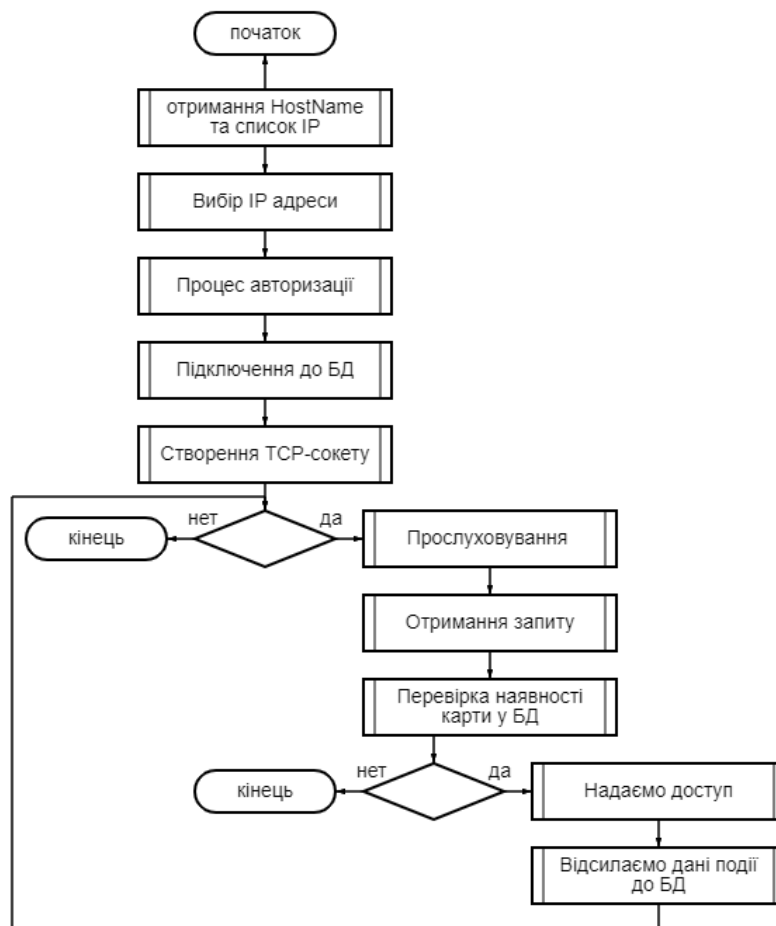


Рисунок 3.24 – Зображення алгоритму роботи методу підключення

Наступним кроком є формування структури реляційної бази даних та її таблиць зберігання й сортування. За допомогою цієї системи зберігатиметься список обов’язкових та додаткових даних, потрібних для працездатності системи в цілому.

Як було розглянуто раніше, основою маніпуляції інформацією між таблицями та пристроєм зчитування в проєкті системи контролю та управління доступом є реляційна система управління БД MySQL.

Для початку роботи з блоком зберігання встановлюється збірка веб-серверу XAMPP. З нею набагато зручніше буде працювати із даними, адже пакет містить потрібну MySQL та веб-інтерфейс PHPMyAdmin, що дозволяє адмініструвати вказану СУДБ.

У панелі контролю збірки обираються порожні порти, за якими працюватиме СУБД, після вмикається режим адміністрування, що переносить до панелі керування PHPMyAdmin.



Рисунок 3.25 – Зображення панелі налаштування програми «Хампрр»

Методом декларативної мови програмування SQL створюється база даних «db\_acs». В базі даних створюються елементи таблиці: «t\_contact», «t\_employ», «t\_log», «t\_position», «t\_terminal», «t\_uid», що умовно поділяються на 2 основні рівня.

- 1) Рівень внесення логуювання даних співробітника (Employee Layer);
- 2) Рівень базової ідентифікаційної інформації, на яку спирається алгоритм доступу до пропуску (Entrance Layer).

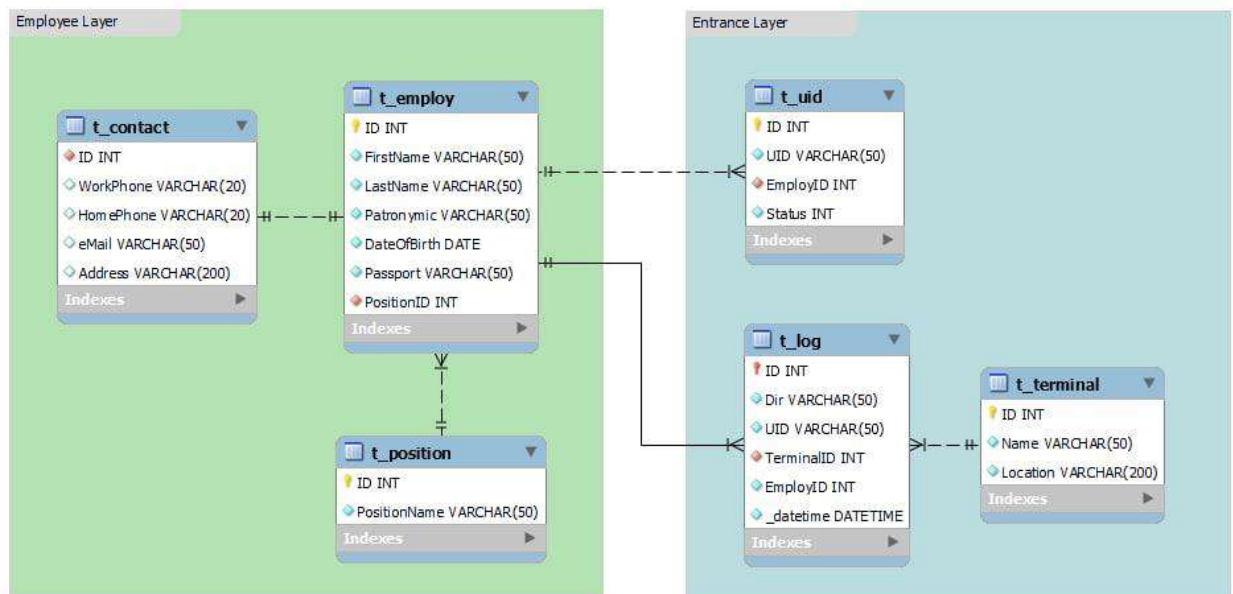


Рисунок 3.26 – Зображення структурної схеми бази даних

Спочатку розглянемо рівень Entrance Layer, що має відповідати за можливість контролю подій входу та виходу, надавати або забороняти ті чи інші дії на пункті пропуску виконавчого пристрою. Із цією шарою працює програма-обробник TCP\_Server, що описувався вище в цьому розділі.

Рівень працює із трьома таблицями: «t\_log», «t\_terminal» та «t\_uid».

- Таблиця «t\_uid» містить 3 осередки: «UID», «EmployID», «Status»
  - 1) «UID»- зберігає ідентифікаційні дані радіочастотних міток на які спирається алгоритм порівняння під час дозволу пропуску;
  - 2) «EmployID»- зберігає номер користувача картою в таблиці;
  - 3) «Status»- зберігає статус активності мітки.

#### Лістинг 3.16 – Створення таблиці «t\_uid»

```
CREATE TABLE `t_uid` (
  `ID` int(11) NOT NULL,
  `UID` varchar(50) NOT NULL,
  `EmployID` int(11) NOT NULL,
  `Status` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Таблиця «t\_log» містить 5 основних осередків: «Dir», «UID», «TerminalID», «EmployID», «\_datetime».
  - 1) «Dir»- зберігає інформацію логування щодо напрямку руху за пунктом пропуску виконавчого пристрою;
  - 2) «UID»- зберігає інформацію логування подій із належним до власника ідентифікаційним номером;
  - 3) «TerminalID»- зберігає логування номеру терміналу виконавчого пристрою;
  - 4) «EmployID»- зберігає логування номеру користувача картою в таблиці;
  - 5) «\_datetime»- зберігає дату та час подій, що відбулися на терміналі.

#### Лістинг 3.17 – Створення таблиці «t\_log»

```
CREATE TABLE `t_log` (
  `ID` int(11) NOT NULL,
  `Dir` varchar(50) NOT NULL,
  `UID` varchar(50) NOT NULL,
  `TerminalID` int(11) NOT NULL,
  `EmployID` int(11) NOT NULL,
  `_datetime` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Таблиця «t\_terminal» містить 2 основних осередки: «Name», «Location».

- 1) «Name»- зберігає ім'я терміналу в системі терміналів;
- 2) «Location»- зберігає місцезнаходження або адресу фізичного терміналу в системі терміналів.

### Лістинг 3.18 – Створення таблиці «t\_terminal»

```
CREATE TABLE `t_terminal` (
  `ID` int(11) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Location` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Рівень бази даних Employee Layer здебільшого задіяний у формуванні та зберіганні даних до пристрою внесення даних нових користувачів і працівників, тим не менш, цей рівень належить до бази даних «db\_acs», отже буде описаний в цьому ж підрозділі.

Рівень працює теж із трьома таблицями: «t\_employ», «t\_position», «t\_contact».

- Таблиця «t\_employ» містить 6 осередків: «FirstName», «LastName», «Patronymic», «DateOfBirth», «Passport», «PositionID».

- 1) «FirstName»- зберігає ім'я співробітників;
- 2) «LastName»- зберігає прізвища співробітників;
- 3) «Patronymic»- зберігає ім'я по-батькові співробітників;
- 4) «DateOfBirth»- зберігає дату народження співробітників;
- 5) «Passport»- зберігає серію та дані паспортів співробітників;
- 6) «PositionID»- зберігає номери посади співробітників.

### Лістинг 3.19 – Створення таблиці «t\_employ»

```
CREATE TABLE `t_employ` (
  `ID` int(11) NOT NULL,
  `FirstName` varchar(50) NOT NULL,
  `LastName` varchar(50) NOT NULL,
  `Patronymic` varchar(50) NOT NULL,
  `DateOfBirth` date NOT NULL,
  `Passport` varchar(50) NOT NULL,
  `PositionID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Таблиця «t\_contact» містить 4 основних осередки: «WorkPhone», «HomePhone», «eMail», «Address».
- 1) «WorkPhone»- зберігає робочі телефони співробітників;
  - 2) «HomePhone»- зберігає домашні телефони співробітників;
  - 3) «eMail»- зберігає електронні адреси співробітників;
  - 4) «Address»- зберігає адреси співробітників.

#### Лістинг 3.20 – Створення таблиці «t\_contact»

```
CREATE TABLE `t_contact` (
  `ID` int(11) NOT NULL,
  `WorkPhone` varchar(20) DEFAULT NULL,
  `HomePhone` varchar(20) DEFAULT NULL,
  `eMail` varchar(50) DEFAULT NULL,
  `Address` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Таблиця «t\_position» містить 1 основний осередок: «PositionName». Він зберігає назви посад співробітників.

#### Лістинг 3.21 – Створення таблиці «t\_position»

```
CREATE TABLE `t_position` (
  `ID` int(11) NOT NULL,
  `PositionName` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

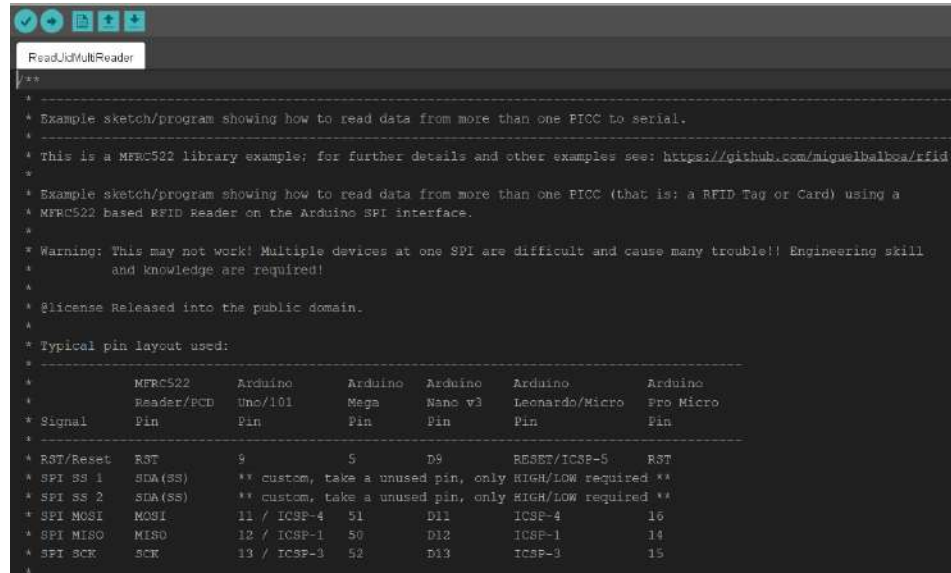
### 3.6 Програмування пристрою внесення даних

Для програмування пристрою внесення даних було обрано використовувати інтегроване середовище розробки «Arduino IDE» через нескладність роботи з ним.

В користі робочого програмного коду було обрано готовий варіант, що сконструйований за допомогою інтегрованої в проект бібліотеки радіо ідентифікаційної мікросхеми MFRC522 мовою програмування C. Програма бібліотека була завантажена за допомогою відкритого посилання.

В користі апаратної платформи було обрано платформу розробника «Arduino», що побудована на мікроконтролері ATmega328.

Назва проекту «ReadUidMultiReader». Його функціонал забезпечує зчитування даних ідентифікаційних даних транспондерів, що потрібно внести в таблицю бази даних, та, саме, відправку зчитаних даних за допомогою послідовного порту на робочу машину, де розгорнуто сервер.



```

ReadUidMultiReader
/**
 * Example sketch/program showing how to read data from more than one PICC to serial.
 *
 * This is a MFRC522 library example; for further details and other examples see: https://github.com/miguelbalboa/rfid
 *
 * Example sketch/program showing how to read data from more than one PICC (that is: a RFID Tag or Card) using a
 * MFRC522 based RFID Reader on the Arduino SPI interface.
 *
 * Warning: This may not work! Multiple devices at one SPI are difficult and cause many trouble!! Engineering skill
 * and knowledge are required!
 *
 * @license Released into the public domain.
 *
 * Typical pin layout used:
 *
 *
 * MFRC522   Arduino   Arduino   Arduino   Arduino   Arduino
 * Reader/PCD Uno/101   Mega      Nano v3   Leonardo/Micro Pro Micro
 * Signal    Pin         Pin       Pin       Pin       Pin
 *
 * RST/Reset  RST        9         5         D9        RESET/ICSP-5  RST
 * SPI SS 1   SDA(SS)    ** custom, take a unused pin, only HIGH/LOW required **
 * SPI SS 2   SDA(SS)    ** custom, take a unused pin, only HIGH/LOW required **
 * SPI MOSI   MOSI       11 / ICSP-4  51       D11       ICSP-4        16
 * SPI MISO   MISO       12 / ICSP-1  50       D12       ICSP-1        14
 * SPI SCK    SCK        13 / ICSP-3  52       D13       ICSP-3        15

```

Рисунок 3.27 – Зображення заголовку програми відправки даних транспондерів

По-перше, підключаються заголовочні файли встановлених функціональних бібліотек, що забезпечують роботу вмонтованих функцій. Бібліотека «MFRC522» та «SPI». Перша дозволяє користуватися командами спілкування с мікросхемою, друга ініціалізує налаштування роботи з SPI-протоколом комунікації.

Лістинг 3.22 – Додавання бібліотек на початку проекту

```

#include <SPI.h>
#include <MFRC522.h>

```

У циклі Main ініціалізується мікросхема зчитування та процес передачі даних за UART-протоколом послідовного порту.

Лістинг 3.23 – Ініціалізація послідовного порту

```

Serial.begin(9600);

```

```
while (!Serial);
SPI.begin();
```

В без кінцевому циклі While перевіряється наявність піднесення картки, проходить алгоритм запобігання зіткнення карт, зчитується сектор, що зберігає ідентифікаційний номер транспондери та відправляється за допомогою послідовного порту. Після цього вимикається та чекає наступного піднесення.

Лістинг 3.24 – Блок без кінцевого циклу

```
if(mfrc522[reader].PICC_IsNewCardPresent() &&
mfrc522[reader].PICC_ReadCardSerial()) {
Serial.print(F("In hex:"));
dump_byte_array(mfrc522[reader].uid.uidByte, mfrc522[reader].uid.size);
Serial.println();
mfrc522[reader].PICC_HaltA();
mfrc522[reader].PCD_StopCrypto1();
```

Передача унікального номеру ідентифікації відбувається у шістнадцятковому форматі чисел по чотири байти в масиві. Кожен байт може приймати значення від 0 до 255 одиниць. Для цього створена окрема функція.

Лістинг 3.25 – Функція прийому байтових даних і створення масиву

```
void dump_byte_array(byte *buffer, byte bufferSize) {
for (byte i = 0; i < bufferSize; i++) {
Serial.print(buffer[i] < 0x10 ? " 0" : " ");
Serial.print(buffer[i], HEX);
```

### 3.7 Програмування додатку роботи з даними співробітників

Невід’ємною складовою зручної та структурно-закінченої системи контролю та управління доступом є гарно сконструйована панель керування даними. Їх внесення, видалення, зміна, корегування та експорт повинні створюватися без потреби втручання до змін програмного коду на низькому рівні системи, адже метою встановлення подібних приладів є полегшення контролю та керування доступом а не ускладнення цих процесів.



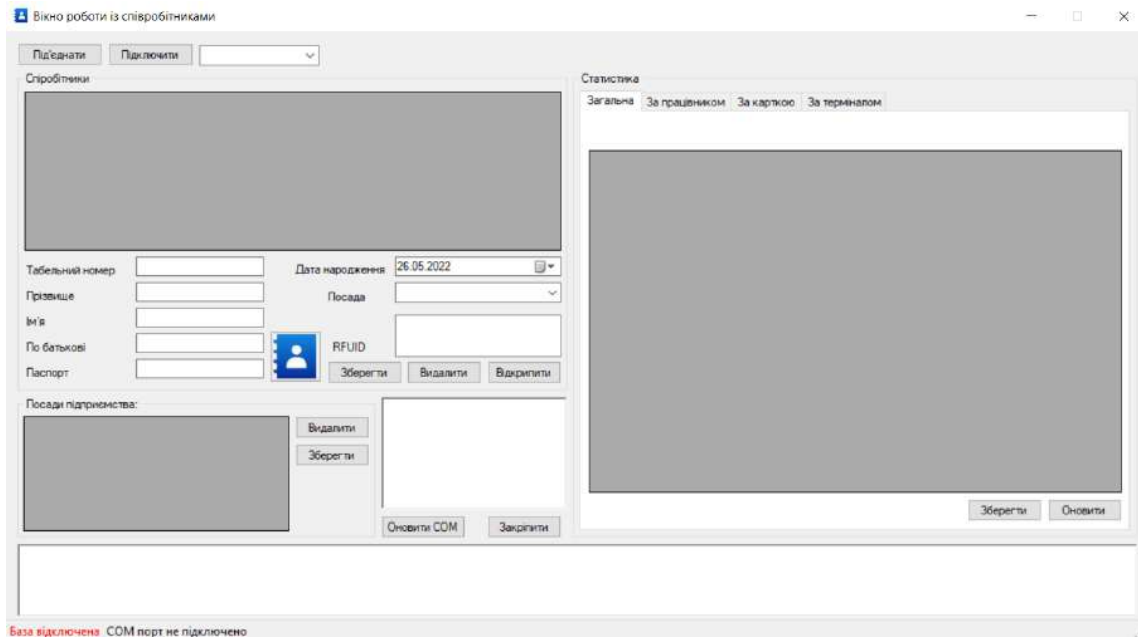


Рисунок 3.28 – Зображення вікна керування даними співробітників

Основою написання додатку в розгалуженій СКУД обрано те ж програмне забезпечення від компанії «Microsoft» що й у випадку програми серверної частини- «Visual Studio». Програмування відбувається мовою С#.

Особливості мови й середовища розробки в порівнянні з іншими мають перевагу. Велика кількість вбудованих інструментів для роботи із вводом та виводом даних із додатку, інструменти, що дозволяють працювати із конструюванням GUI (Graphic User Interface)- форм графічного інтерфейсу, достатніх за складністю для задовільнення усіх поставлених до проекту завдань.

Основними потребами від додатку роботи з даними співробітників є зручність внесення та зміни ідентифікаційних даних та моніторингу даних логуювання подій, що відбуваються на виконавчому пристрої системи. Додаток має містити список співробітників (ПІБ), номери «прив'язаних» до них транспондерів, дати народження, номери документів підтвердження особи, посади, контактні дані. Також, інтерфейс має дозволяти отримувати контролюючому дані статистики за кожним співробітником, особу якого внесено до бази даних СКУД.

В проєкті використовуються такі функціональні бібліотеки як:

## Лістинг 3.26 – Додавання бібліотек на початку проекту

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data;
using MySql.Data.MySqlClient;
using System.IO;
using System.IO.Ports;
using iTextSharp.text;
using iTextSharp.text.pdf;

```

Клас `System` означає, що використовується бібліотека `System` у проекті. Це дає деякі корисні класи та функції, такі як клас `Console` або функція/метод `WriteLine`.

Клас `System.Collections.Generic` забезпечує загальну реалізацію стандартної структури даних, як-от пов'язані списки, стеки, черги та словники.

Клас `System.ComponentModel` надає інтерфейси, які використовуються для реалізації поведінки компонентів під час виконання та проектування.

Клас `System.Data` дозволяє об'єкту реалізувати `DataAdapter` і представляє набір методів і властивостей, пов'язаних із дією, які використовуються для заповнення та оновлення набору даних та оновлення джерела даних.

Клас `System.Drawings` відкриває доступ до функціональних можливостей роботи з графічним інтерфейсом.

Клас `System.Linq` надає різні типи класів і методів, які підтримують запити LINQ.

Клас `System.Text` містить класи, які представляють кодування символів ASCII та Unicode. Абстрактні базові класи для перетворення блоків символів у блоки байтів та з них і допоміжний клас, який маніпулює та форматує об'єкти `String` без створення проміжних екземплярів `String`.

Клас `System.Threading.Tasks` надає типи, які спрощують роботу з написанням паралельного та асинхронного коду.

Клас `MySql.Data.MySQL` надає серію класів у `MySQL Connector/Net`. Весь зв'язок між програмою `C#` і сервером `MySQL` маршрутизується через об'єкт `MySqlConnection`.

Далі потрібно встановлення бібліотеки `MySql.Data.MySqlClient`, за допомогою якої програма `Dot Net` матиме змогу зв'язку із сервером `MySQL`. Після встановлення бібліотеки з офіційного сайту `MySQL`, вона так само додається до проекту за допомогою функції `using`.

Класи `System.IO` та `System.IO.Ports` - ці класи можна використовувати для читання та запису даних у файли або потоки даних.

Класи `iTextSharp.text` та `TextSharp.text.pdf` — це розширені бібліотеки інструментів, які використовуються для створення складних звітів у форматі `PDF`.

Процес створення додатку полягає у будівництві конструктора форми та додаванню їй окремих структурних елементів, таких як: поля, кнопки та вікна. Кожен елемент має свою окрему функцію-обробник, алгоритм роботи якої будується індивідуально за потребами, та панель налаштувань і властивостей.

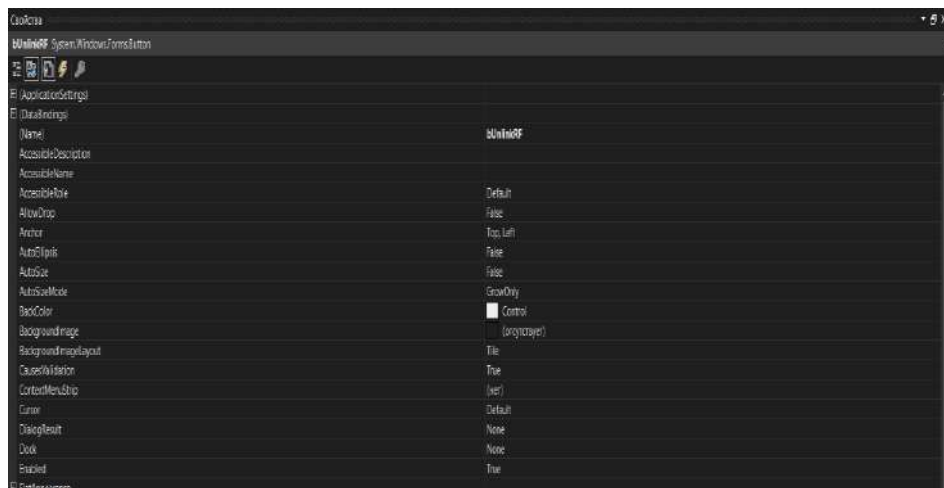


Рисунок 3.29 – Зображення панелі налаштування кнопки в програмі «Visual Studio 2022»

Далі зображено у виді блок-схем алгоритм найважливіших функціональних елементів- кнопок (button). При натисканні кнопки викликається функція-обробник переривання, що виконує створений для неї алгоритм обробки.

Алгоритм роботи кнопки підключення «bConnection» до бази даних «db\_acs» та відключення. Передача даних з послідовного порту відбувається в асинхронному режимі за допомогою метода «Handler».



Рисунок 3.30 – Зображення алгоритму роботи методу підключення

Алгоритм роботи кнопки закріплення ідентифікаційного номеру з транспондери за окремим співробітником. За допомогою цього алгоритму здійснюється додавання нового ідентифікаційного номеру співробітнику, зміна вже існуючого та видалення вже безпосередньо існуючого.

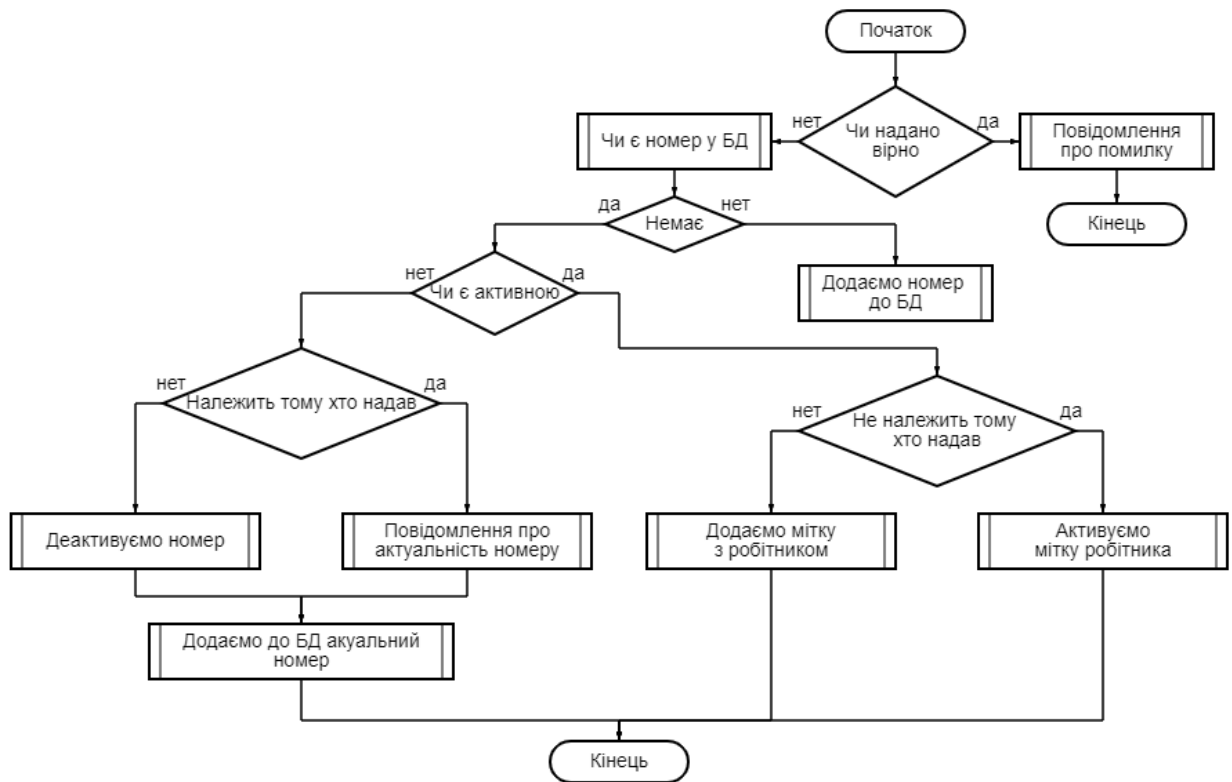


Рисунок 3.31 – Зображення алгоритму роботи кнопки закріплення ідентифікаційного номеру з транспогдеру

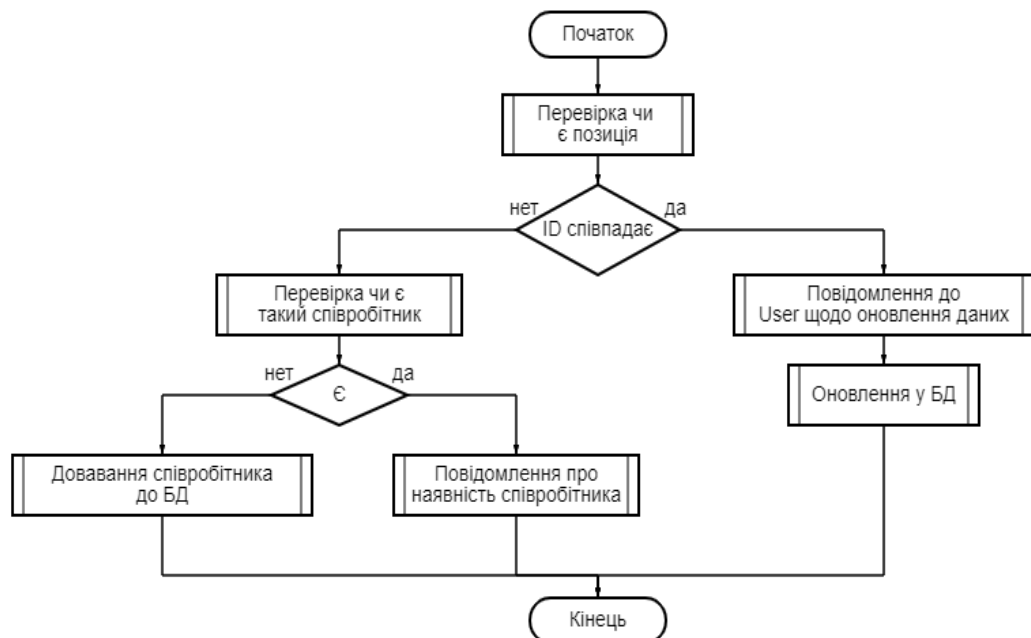


Рисунок 3.32 – Зображення алгоритму роботи кнопки збереження даних співробітників

Алгоритм роботи кнопки збереження даних співробітників. За допомогою алгоритму є можливість додавання прізвищ імен та по-батькові, табельних номерів, дат народження, посад та даних документів підтвердження осіб.

Додавання контактних даних є функцією, за допомогою якої є можливість зберігати номери телефонів та електронні адреси й адреси місця проживання співробітників. Для цього створено окрему форму «контактні дані співробітника».

Рисунок 3.33 – Зображення вікна додавання комунікаційних даних співробітників

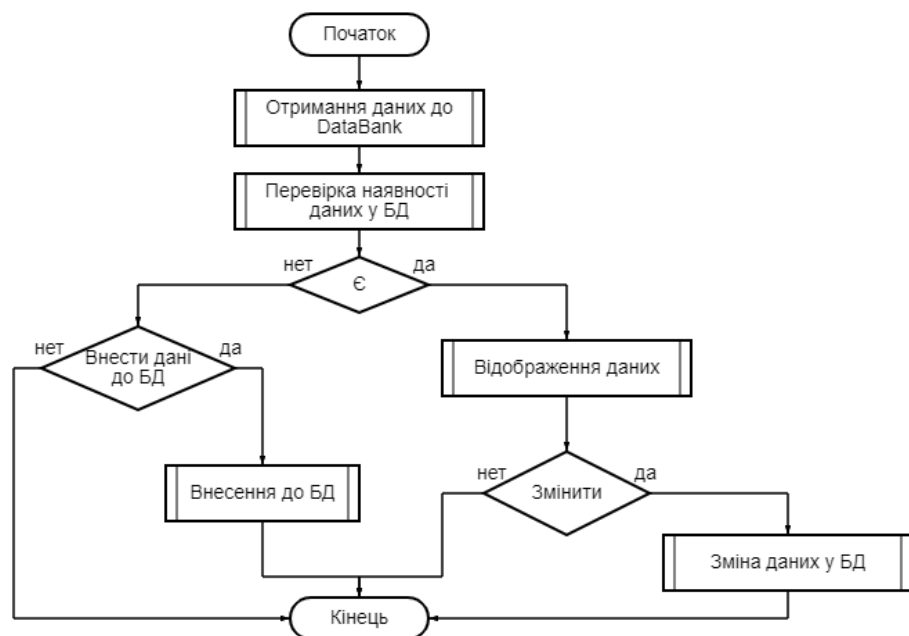


Рисунок 3.34 – Зображення алгоритму роботи кнопки переносу до панелі додавання комунікаційних даних

За допомогою спеціальної кнопки елемента «button» потрапляємо в окрему форму редагування даних. Обмін між двома формами відбувається за допомогою методів створеного класу «DataBank», що надає функції буферної зони для можливості передачі даних.

### 3.8 Тестування та налагодження СКУД

Щоб привести систему контролю управління доступом до робочого стану треба зробити налаштування вузлів та їх фізичне підключення у єдину структуру, що функціонуватиме одночасно та в той же час незалежно.

Хоч і треба прагнути зменшення кількості утручань у роботу системи, первинний запуск системи потребує налаштування, як програмного так і апаратно- функціонального.

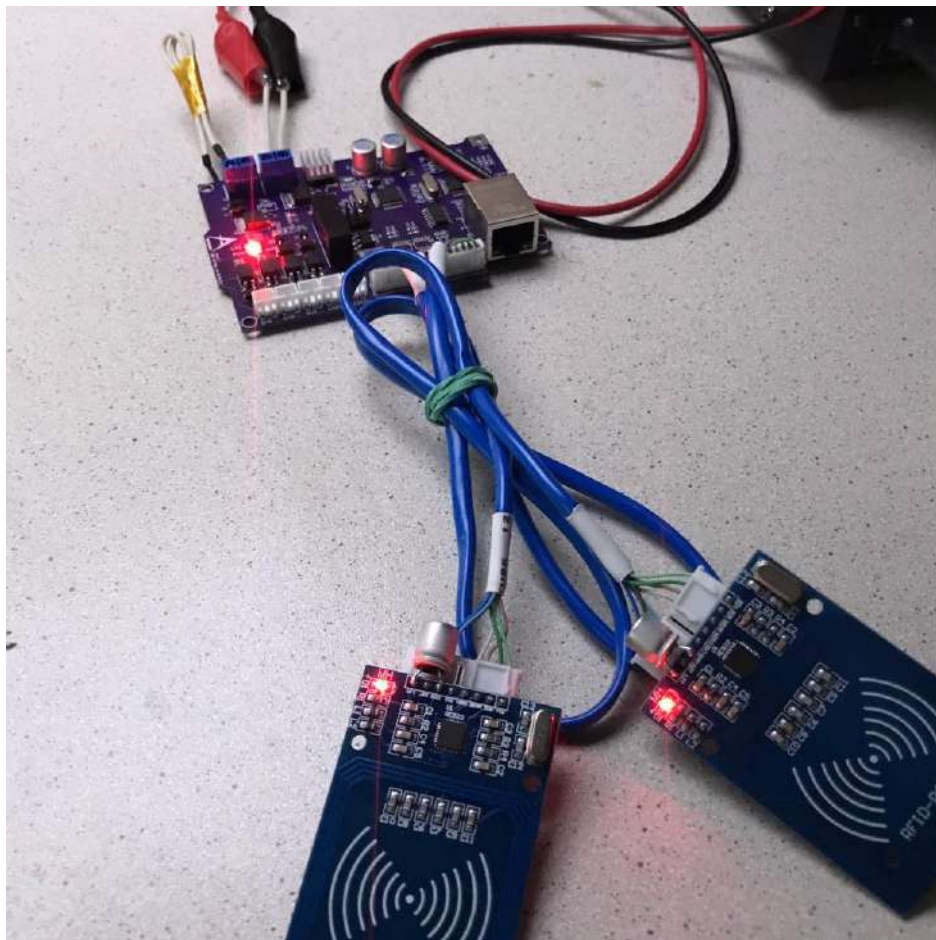


Рисунок 3.35 – Зображення виконавчого пристрою СКУД із приладами зчитування

По-перше, до програматора ST-link v.2.0 треба підключити контролер (виконавчий пристрій) СКУД за допомогою порту передачі даних SWD. В програмному коді треба встановити актуальну IP-адресу машини, за допомогою якої розгорнуто сервер, адресу маршрутизатора мережі та адресу контролера виконавчого пристрою. За допомогою цих даних розгалужені вузли системи зможуть отримати інформацію щодо правил підключення один до одного в суспільній мережі.

```
#define DEST_IP           {192,168,31,185}
#define LOCAL_IP         {192, 168, 31, 103}
#define SUBNET           {255, 255, 255, 0}
#define GATEWAY          {192, 168, 31, 1}
#define DNS              {0, 0, 0, 0}
#define DHCP             NETINFO_DHCP//NETINFO_STATIC
```

Рисунок 3.36 – Зображення фрагменту програмного коду

Знайшовши кнопку компіляції проекту та його завантаження імпортуємо проект, програмуючи мікроконтролер СКУД.

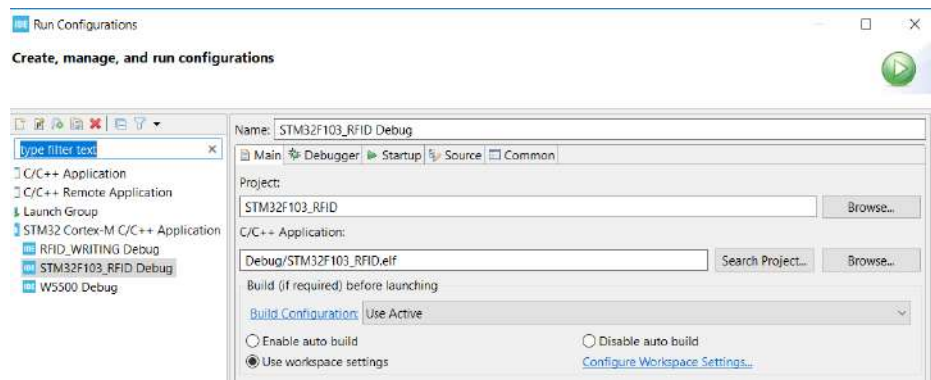


Рисунок 3.37 – Зображення вікна завантаження програмного коду до мікроконтролера

По-друге, відкривши панель керування програми «ХАМРР», треба запустити роботу бази даних «MySQL» та допоміжної програми «Apache». За допомогою функціональної кнопки «Admin» розпочинається робота серверу «MySQL».





Рисунок 3.38 – Зображення панелі програми «ХАМРР»

Наступна дія- розгортання консольного додатку, що оброблятиме потоки надходження інформації від виконавчого пристрою та, навпаки, відправлятиме дані, оброблені додатком керування сервером та базою даних до початкового вузла. Для цього запускається «.exe» файл проекту. Пишемо ідентифікаційні дані для з'єднання із базою даних.

```
C:\Users\User\Desktop\TCP_Server.exe
Local Machine's Host Name: WIN-0379GHARNB8
IP Address 0: 192.168.31.185
IP Address 1: fe80::fc46:d49f:3a55:581a%16
Server Started at 192.168.31.185:8080
Input login: _
```

Рисунок 3.39 – Зображення панелі консольного додатка-обробника серверної частини

Щоб мати змогу вносити, видаляти, корегувати та отримувати дані з таблиць бази даних було створено графічний додаток «WinHuman». Розгортається він також за допомогою розширення додатку «.exe».

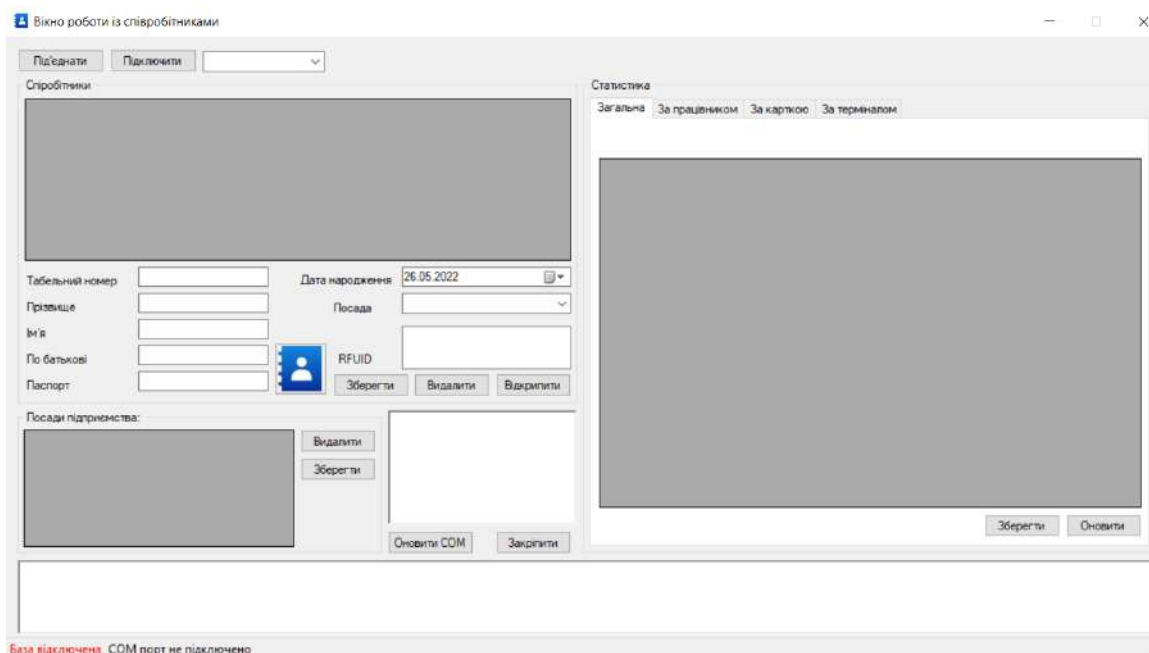


Рисунок 3.40 – Зображення вікна роботи із співробітниками

Із програмним налаштуванням проекту закінчено. Робиться це одноразово, саме під час першого запуску системи.

Апаратне тестування потребує підключення контролеру СКУД до електричного живлення постійного току, напругою 12 В. Тестування проходитиме з електронним навантаженням на виході виконавчого пристрою замість електро-магнітної системи. Струм навантаження виставляється біля 1000 Ма, що повністю емулює підключення реального навантаження.

Для роботи приладу треба забезпечити достатній рівень струму, щоб можна було жити як електронне навантаження на виході контролера, так і логічну частину контролера, без зменшення напруги, що її живить. Для того, щоб дізнатись потрібні технічні дані блоку живлення- скористаємось лабораторним джерелом, виставивши напругу на виході біля 12 В.

Підключимо контролер СКУД до лабораторного блоку живлення без задіяння електронного навантаження. Струм споживання контролера для забезпечення своїх потреб 108 мА. За законом Ома, потужність рівняється добутку струму з напругою:  $W=U \cdot I$ ,  $12 \cdot 0.108 = 1.296$ , що, приблизно, дорівнює 1,3 Вт потужності. Спираючись на це- блок живлення треба обирати таких,

який зможе забезпечити живлення СКУД та навантаження, а це у купі близько 1.5 А при 12 В.



Рисунок 3.41 – Зображення лабораторного джерела живлення

За допомогою спеціалізованого кабелю Ethernet з'єднання із фізичними роз'ємами типу RJ45, підключаємо контролер СКУД до маршрутизатора, вхід локальної (LAN- local area network) мережі маршрутизатора.

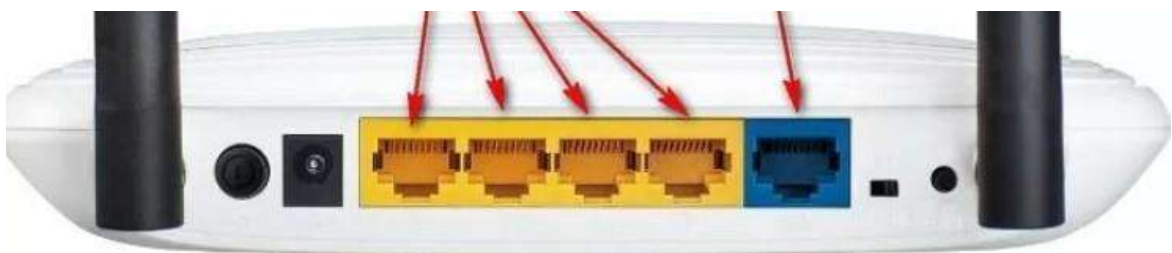


Рисунок 3.42 – Зображення панелі виходів RJ45 на мережевому маршрутизаторі

Машину, за допомогою якої розгорнуто серверну частино теж треба підключити в систему локальної мережі для забезпечення суспільного зв'язку на одному рівні. Фізичне підключення виконано та система готова до роботи.

Щоб протестувати програмний алгоритм системи та коректність виконання команд, треба скористатись створеним інтерфейсом керування. Відкривається файл додатку «WinHuman.exe», за допомогою функції «Під'єднати» додаток під'єднується до бази даних та вносяться гіпотетичні дані співробітника, а саме: його ПІБ, паспортних даних, дати народження, посади та табельного номера. Перед цим вносяться список посад підприємства.

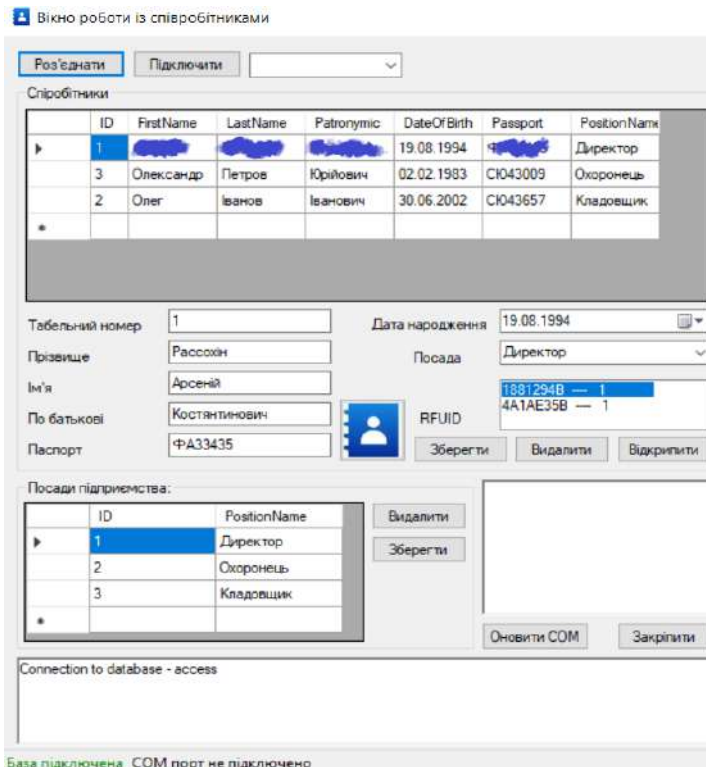


Рисунок 3.43 – Зображення панелі додавання співробітників

Після внесення даних особи, за допомогою функціональної кнопки «Підключити» система під'єднується до пристрою внесення даних. Пристрій встановлює з'єднання із машиною, за допомогою якої розгорнуто сервер, використовуючи послідовний порт, отже треба підключити його за допомогою USB порту. Далі скануємо ідентифікаційну мітку-транспондер, закріплюємо за особою співробітника, що додана.

Паралельно, усі необхідні дані повинні бути автоматично внесені до таблиць бази даних «db\_acs».

				ID	UID	EmployID	Status			
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	1881294B	1	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	4A1AE35B	1	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	BA94DB5B	3	1

Рисунок 3.44 – Зображення панелі ідентифікаційних номерів в програмі «phpMyAdmin»

Прикладаємо внесений транспондер до зчитувача входу або виходу-отримуємо результат. Вихід контролера СКУД переходить до активного стану та видає напругу на підключене до цього моменту електронне навантаження. Дисплей навантаження, що емулює електричний магніт, демонструє наявність потрібних характеристик, що надає контролер, потрібних для роботи замка закриття.



Рисунок 3.45 – Зображення показників електронного навантаження

### 3.9 Висновки за розділом

Результатом виконання розділу є проектування апаратної та програмної частин системи контролю та управління доступом. Також, було створено додаток керування внесенням, видаленням та корегуванням даних співробітників в систему бази даних. Наприкінці, було налаштовано та протестовано систему в робочому вигляді.

## ВИСНОВКИ

У результаті виконання роботи дипломного проекту було розглянуто предметну область систем контролю та керування доступом, види, методи та способи їх використання на практиці. Оцінено актуальне положення систем контролю, рівень глибини, популярності та доцільності їх використання в сучасному світі. Установлено доцільність створення клієнт-серверної системи, спираючись та існуючі аналоги та потреби в комплексах охорони та слідкування складських приміщень.

Розроблено прилад виконавчого пристрою на базі одно кристального ЕОП, з додаванням фізичного контролера безпечного мережевого зв'язку. Інтегровано систему контролю фізичного доступу, за допомогою керування замком електромагнітного типу.

Розроблено програмний алгоритм керування та з'єднання серверної частини, що була розгорнута за допомогою ІВМ-подібної машини, із вузлом виконавчого пристрою з метою створення можливості комунікації та швидкого обміну даними на великій дистанції.

Розроблено графічний додаток панельного типу для внесення й керування ідентифікаційними даними співробітників складського приміщення з використанням таблиць, розташованих в базі зберігання даних.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Система контролю і управління доступом [Електронний ресурс] / Режим доступу [www. URL: https://uk.wikipedia.org/wiki/Система\\_контролю\\_та\\_управління\\_доступом](https://uk.wikipedia.org/wiki/Система_контролю_та_управління_доступом) - 10.04.22.
2. Системи контролю та управління доступом [Електронний ресурс] / Режим доступу [www. URL: https://www.bezpeka-shop.com/catalog/kontrol\\_dostupa\\_20436/](https://www.bezpeka-shop.com/catalog/kontrol_dostupa_20436/)
3. СКУД. Системи контролю доступу [Електронний ресурс] / Режим доступу [www. URL: https://www.aktivsb.ru/kontrol\\_dostupa/](https://www.aktivsb.ru/kontrol_dostupa/)
4. Access Control [Електронний ресурс] / Режим доступу [www. URL: https://www.techtarget.com/searchsecurity/definition/access-control](https://www.techtarget.com/searchsecurity/definition/access-control)
5. What are the benefits of Access Control Systems? [Електронний ресурс] / Режим доступу [www. URL: https://cie-group.com/how-to-av/videos-and-blogs/access-control-systems](https://cie-group.com/how-to-av/videos-and-blogs/access-control-systems)
6. What is access control? [Електронний ресурс] / Режим доступу [www. URL: https://www.fortinet.com/resources/cyberglossary/access-control](https://www.fortinet.com/resources/cyberglossary/access-control)
7. Мікроконтролери початок. [Електронний ресурс] / Режим доступу [www. URL: https://www.youtube.com/watch?v=\\_YfayPCMLMQ&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=_YfayPCMLMQ&ab_channel=VladimirMedintsev)
8. Вивчаємо STM 32 [Електронний ресурс] / Режим доступу [www. URL: https://www.youtube.com/watch?v=vZVwvEeHkI&t=2s&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=vZVwvEeHkI&t=2s&ab_channel=VladimirMedintsev)
9. STM 32 Доступні операційні системи. [Електронний ресурс] / Режим доступу [www. URL: https://www.youtube.com/watch?v=Da3-xi9GQcw&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=Da3-xi9GQcw&ab_channel=VladimirMedintsev)
10. Структура коду в STM32CubeIDE [Електронний ресурс] / Режим доступу [www. URL:](#)



[https://www.youtube.com/watch?v=l9c2XNeuqJU&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=l9c2XNeuqJU&ab_channel=VladimirMedintsev)

11. STM32 + RC522 Як зберігати та копіювати конфігурацію пристрою. Як працюють картки NXP MIFARE [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=d-kwCaREpBE&t=1160s&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=d-kwCaREpBE&t=1160s&ab_channel=VladimirMedintsev)
12. Інтерфейс SPI з прикладу STM32. Підключення периферії - екрану та пам'яті [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=6ruS8eR\\_Rdc&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=6ruS8eR_Rdc&ab_channel=VladimirMedintsev)
13. STM32 + Ethernet / Як вибрати Ethernet контролер / Налаштування бібліотеки LWIP / LAN8720 / Частина 1. [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=vw-2mkR8xlg&t=3s&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=vw-2mkR8xlg&t=3s&ab_channel=VladimirMedintsev)
14. STM32 + Ethernet / Чіп W5500 від WIZnet / Сокети та проста розповідь про побудову WEB сервера. Частина 2. [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=PTDoDJadyi0&t=1172s&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=PTDoDJadyi0&t=1172s&ab_channel=VladimirMedintsev)
15. Програмування МК STM32. Урок 91. LAN. W5500. HTTP Server. Частина 1 [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=W85Zaysv2PY&ab\\_channel=narodstrem](https://www.youtube.com/watch?v=W85Zaysv2PY&ab_channel=narodstrem)
16. Ethernet для МК. W5500 та ENC28J60. MQTT [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=LwDDEIx63cA&ab\\_channel](https://www.youtube.com/watch?v=LwDDEIx63cA&ab_channel)
17. Від ідеї до готового пристрою. З чого починати опрацювання ідеї майбутнього електронного пристрою? [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=lb4bdQKGns&ab\\_channel=VladimirMedintsev](https://www.youtube.com/watch?v=lb4bdQKGns&ab_channel=VladimirMedintsev)

18. Datasheet STM32F103 [Електронний ресурс] / Режим доступу www. URL: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>
19. IPRAW. [Електронний ресурс] / Режим доступу www. URL: <https://docs.wiznet.io/Product/iEthernet/W5500/Application/ipraw/>
20. Datasheet W5500 [Електронний ресурс] / Режим доступу www. URL: <https://docs.wiznet.io/Product/iEthernet/W5500/datasheet/>
21. Limitation Note [Електронний ресурс] / Режим доступу www. URL: [https://docs.wiznet.io/assets/files/limitation\\_note\\_arp\\_problem\\_in\\_the\\_nlb\\_environment\\_-\\_english\\_0312\\_-b8d367a148e870c05217389964f76f15.pdf](https://docs.wiznet.io/assets/files/limitation_note_arp_problem_in_the_nlb_environment_-_english_0312_-b8d367a148e870c05217389964f76f15.pdf)
22. Захист схем від переполюсування живлення за допомогою N-канального MOSFET [Електронний ресурс] / Режим доступу www. URL: <https://myelectrons.ru/mosfet-reversed-protection/>
23. Вибираємо мікроконтролер разом [Електронний ресурс] / Режим доступу www. URL: <https://habr.com/ru/post/122030/>
24. Ізольовані прийомопередавачі SPI [Електронний ресурс] / Режим доступу www. URL: <https://www.analog.com/ru/product-category/spisolator.html>
25. MAX14850AEE+ Datasheet [Електронний ресурс] / Режим доступу www. URL: <https://www.alldatasheet.com/datasheetpdf/pdf/1366756/MAX14850AEE+.html>
26. K3918 2SK3918 Datasheet [Електронний ресурс] / Режим доступу www. URL: <https://datasheetpdf.com/datasheet/K3918.html>
27. HR911105A Datasheet [Електронний ресурс] / Режим доступу www. URL: [https://datasheet.lcsc.com/szlcsc/Zhongshan-HanRun-Elec-HR911105A\\_C12074.pdf](https://datasheet.lcsc.com/szlcsc/Zhongshan-HanRun-Elec-HR911105A_C12074.pdf)
28. STM32 + CMSIS + STM32CubeIDE [Електронний ресурс] / Режим доступу www. URL: <https://habr.com/ru/post/481478/>
29. CH340 Datasheet [Електронний ресурс] / Режим доступу www. URL: <https://html.alldatasheet.com/htmlpdf/1132602/WCH/CH340/60/1/CH340.html>

30. EM4100 Protocol description. [Електронний ресурс] / Режим доступу www. URL: [https://www.priority1design.com.au/em4100\\_protocol.html](https://www.priority1design.com.au/em4100_protocol.html)
31. Em-Marine - огляд стандарту, як правильно вибрати карти, брелоки, браслети та зчитувачі? [Електронний ресурс] / Режим доступу www. URL: [https://securityrussia.com/blog/em\\_marine.html](https://securityrussia.com/blog/em_marine.html)
32. Біометрія: що це і як вона змінює світ фінансів [Електронний ресурс] / Режим доступу www. URL: <https://fincult.info/article/biometriya-chto-eto-i-kak-ona-menyaet-mir-finansov/>
33. AMS1117-3.3 Datasheet [Електронний ресурс] / Режим доступу www. URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/205691/ADMOS/AMS1117-3.3.html>
34. Уроки C# .NET Windows Forms / #3 - Підключення MySQL та створення бази даних [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=pNTQy48Pafc&list=PL0lO\\_mIqDDFWOMqSKFaLypANf1W7-o87q&index=10&ab\\_channel](https://www.youtube.com/watch?v=pNTQy48Pafc&list=PL0lO_mIqDDFWOMqSKFaLypANf1W7-o87q&index=10&ab_channel)
35. Уроки C# .NET Windows Forms / #3 - Підключення MySQL та створення бази даних [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=pNTQy48Pafc&list=PL0lO\\_mIqDDFWOMqSKFaLypANf1W7-o87q&index=10&ab\\_channel](https://www.youtube.com/watch?v=pNTQy48Pafc&list=PL0lO_mIqDDFWOMqSKFaLypANf1W7-o87q&index=10&ab_channel)
36. Сокети (socket) та клієнт-серверна взаємодія за протоколами TCP та UDP у C# - Вчимо Шарп #18 [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=ZRGgBtUgJKE&ab\\_channel=CODEBLOG](https://www.youtube.com/watch?v=ZRGgBtUgJKE&ab_channel=CODEBLOG)
37. Уроки C# .NET Windows Forms / #4 - Авторизація користувача через базу даних [Електронний ресурс] / Режим доступу www. URL: [https://www.youtube.com/watch?v=eLQAgHexThM&ab\\_channel](https://www.youtube.com/watch?v=eLQAgHexThM&ab_channel)
38. B1205S-2W Datasheet [Електронний ресурс] / Режим доступу www. URL: <https://www.alldatasheet.com/datasheetpdf/pdf/274840/MORNSUN/B1205>

- [S2W.html?gclid=CjwKCAjwjtOTBhAvEiwASG4bCPIWgfavKWT2Uvn8aA0JMAQyEWST56V5yg\\_gvYaheTL7BIkvKPLBNhoCTfwQAvD\\_BwE](https://www.youtube.com/watch?v=eLQAgHexThM/A0JMAQyEWST56V5yg_gvYaheTL7BIkvKPLBNhoCTfwQAvD_BwE)
39. Авторизація користувачів у базі даних. [Електронний ресурс]. Режим доступу: [www](https://www.youtube.com/watch?v=eLQAgHexThM/A0JMAQyEWST56V5yg_gvYaheTL7BIkvKPLBNhoCTfwQAvD_BwE). URL: [https://www.youtube.com/watch?v=eLQAgHexThM/A0JMAQyEWST56V5yg\\_gvYaheTL7BIkvKPLBNhoCTfwQAvD\\_BwE](https://www.youtube.com/watch?v=eLQAgHexThM/A0JMAQyEWST56V5yg_gvYaheTL7BIkvKPLBNhoCTfwQAvD_BwE)
40. Переваги мови С#. [Електронний ресурс]. – Режим доступу: [www](https://forum.itvdn.com/t/urok-1-preimushhestva-yazyka-c/2941/). URL: <https://forum.itvdn.com/t/urok-1-preimushhestva-yazyka-c/2941/>
41. Протоколи TCP та UDP. [Електронний ресурс]. – Режим доступу: [www](https://professorweb.ru/my/csharp/web/level1/1_4.php/). URL: [https://professorweb.ru/my/csharp/web/level1/1\\_4.php/](https://professorweb.ru/my/csharp/web/level1/1_4.php/)
42. Класи та об'єкти. [Електронний ресурс]. – Режим доступу: [www](https://metanit.com/sharp/tutorial/3.1.php). URL: <https://metanit.com/sharp/tutorial/3.1.php>
43. Сокети та клієнт-серверна взаємодія. [Електронний ресурс]. – Режим доступу: [www](https://www.youtube.com/watch?v=ZRGgBtUgJKE&ab_channel=CODEBLOG/). URL: [https://www.youtube.com/watch?v=ZRGgBtUgJKE&ab\\_channel=CODEBLOG/](https://www.youtube.com/watch?v=ZRGgBtUgJKE&ab_channel=CODEBLOG/)
44. Пам'ятка по SQL. [Електронний ресурс]. – Режим доступу: [www](https://habr.com/ru/post/564390/). URL: <https://habr.com/ru/post/564390/>