



Я, Волоніхін Вадим Денисович як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

16 червня 2025 р.

A handwritten signature in black ink, consisting of several overlapping loops and a long horizontal stroke extending to the right.

Вадим ВОЛОНІХІН

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет \_\_\_\_\_ АКТ \_\_\_\_\_  
Кафедра \_\_\_\_\_ КІТАР \_\_\_\_\_  
Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_  
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ Освітньо-професійна \_\_\_\_\_  
Освітня програма Автоматизація та комп'ютерно-інтегровані технології \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:  
Зав. кафедри КІТАР \_\_\_\_\_ (підпис)  
« 28 » квітня 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві \_\_\_\_\_ Волоніхіну Вадиму Денисовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи контролю міського трафіку на базі мікроконтролера stm32

затверджена наказом по університету від 19.05.2025 р. № 390 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 20.06.2025 р.

3. Вихідні дані до роботи \_\_\_\_\_

Середовище розробки CUBE IDE

Середовище автоматизації проектування електроніки Easy EDA

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

4.1 Вступ;

4.2 Аналіз літератури за темою;

4.3 Вибір та тестування мікроконтролера;

4.4 Розроблення друкованої плати;

4.5 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Графічний матеріал у вигляді презентації, файл формату \*.pptx, 12 арк. ф. А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вступ	28.04-04.05.2025	виконано
2	Аналіз технічного завдання. Аналіз публікацій	05.05-10.05.2025	виконано
3	Вибір мікроконтролера	11.05-17.05.2025	виконано
4	Розроблення друкованої плати	18.05-31.05.2025	виконано
5	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	01.06-03.06.2025	виконано
6	Оформлення пояснювальної записки	04.06-08.06.2025	виконано
7	Подання роботи на рецензію	09.06-11.06.2025	виконано
8	Подання роботи на підпис зав. кафедри	12.06-14.06.2025	виконано
9	Подання кваліфікаційної роботи в ЕК	15.06.2025	виконано

Дата видачі завдання 28 квітня 2025 р.

Здобувач \_\_\_\_\_ Вадим ВОЛОНІХІН  
(підпис) (власне ім'я, прізвище)

Керівник роботи \_\_\_\_\_ доц. Наталія ДЕМСЬКА  
(підпис) (посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка: 60 с., 1 табл., 17 рис., 36 джерел.

### МІКРОКОНТРОЛЛЕР STM32, АВТОМАТИЗАЦІЯ ТРАФІК

Мета роботи – підвищення ефективності управління міським трафіком за допомогою автоматизації світлофорів на базі мікроконтролера STM32.

Об'єкт розробки – процес автоматизації управління міським трафіком шляхом інтеграції системи контролю світлофорів.

Предмет розробки – апаратно-програмний комплекс для управління світлофорами, що включає програмування мікроконтролера STM32, розробку алгоритмів регулювання та дизайн друкованої плати для інтеграції компонентів системи.

У даній кваліфікаційній роботі було детально розглянуто сучасні підходи до автоматизації управління світлофорами та аналіз існуючих рішень для контролю міського трафіку. Було проведено огляд апаратних засобів та програмних методів, що використовуються в подібних системах, визначено їх переваги та недоліки, а також досліджено можливості їх застосування у реальних умовах міського середовища.

У процесі роботи розроблено структурну схему системи управління світлофорами, що стала базою для вибору необхідних апаратних компонентів, серед яких мікроконтролер STM32, датчики руху та засоби зв'язку між компонентами системи. На основі розробленої схеми було створено детальну схему підключення елементів, яка забезпечує коректну взаємодію апаратних засобів та гарантує стабільність роботи системи в реальному часі.

Крім того, було розроблено алгоритм роботи системи, що описує принципи функціонування елементів, регулювання режимів роботи світлофорів та управління потоками транспорту. Реалізація алгоритму дозволяє ефективно адаптувати систему до змінних умов дорожнього руху.

Даний підхід дозволяє інтегрувати розроблену систему в існуючі міські інфраструктури, сприяючи оптимізації транспортного потоку, зниженню затримок на перехрестях та підвищенню загальної ефективності управління міським трафіком.

## ABSTRACT

Explanatory note: 60 p., 1 tables, 17 figures, 36 sources.

### STM32 MICROCONTROLLER, TRAFFIC AUTOMATION.

The purpose of the work is to develop an automated urban traffic control system based on the STM32 microcontroller. The goal is to improve traffic light management and optimize the flow of vehicles in the city.

The object of development is the process of automating urban traffic control by integrating a traffic light system.

The subject of development is a hardware-software system for managing traffic lights. This includes programming the STM32 microcontroller, creating control algorithms, and designing a printed circuit board.

In this work, existing methods for automating traffic light control were examined, along with how these methods are applied in real urban environments. Different hardware and software approaches were analyzed, their advantages and disadvantages identified, and their practical uses investigated.

During the work, a structural diagram of the traffic light control system was created. This diagram helped in selecting the necessary hardware components—such as the STM32 microcontroller, motion sensors, and communication modules. Based on this diagram, a connection diagram was also developed to show exactly how the components should be connected for proper operation.

In addition, an operational algorithm was developed. This algorithm clearly explains how the system works, how it switches between different modes, and how the components interact.

## ЗМІСТ

Перелік умовних скорочень .....	7
Вступ.....	9
1 Аналіз літератури за темою.....	11
1.1 Аналіз стандартів DIN VDE.....	11
1.2 Аналіз правил RiLSA .....	13
1.3 Аналіз вимог до створення приладу автоматизованого керування.....	16
1.4 Допоміжні інструменти та пристрої для автоматизації та контролю трафіку.....	22
2 Вибір та налаштування мікроконтролера.....	25
2.1 Вибір мікроконтролера.....	25
2.2 Вибір програмного фреймворку.....	29
2.3 Вибір середи розробки.....	32
2.4 Налаштування функцій мікроконтролера.....	35
3 Розроблення функціонального макету.....	41
3.1 Вибір середи проектування електроніки.....	41
3.2 Перелік та опис використаних компонентів.....	45
3.3 Тестування критичних протоколів.....	48
3.4 Трасування друкованої плати.....	53
3.5 Охорона праці під час розробки, налагодження й експлуатації автоматизованої системи.....	56
Висновки.....	58
Перелік джерел посилання.....	68
Додаток А Апробація результатів роботи.....	61
Додаток Б Лістинг матеріалів друкованої плати (BILL OF MATERIALS).....	62
Додаток В Лістинг коду для CUBE IDE.....	64
Додаток Г Демонстраційний матеріал.....	83

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

EMC – електромагнітна сумісність;

ККД – коефіцієнт корисної дії;

ЦП – центральний процесор;

3D – Three-Dimensional;

BOM – Bill of Materials (перелік усіх компонентів і матеріалів, потрібних для виробництва плати, з зазначенням кількості, артикулів та постачальників);

DIN – Deutsches Institut für Normung (німецький інститут стандартизації);

DRC – Design Rule Check (автоматизована перевірка відповідності топології плати та розміщення компонентів встановленим проектним правилам (мінімальні відстані, ширина доріжок тощо);

EDA – Electronic Design Automation (категорія програмних інструментів для автоматизації процесів створення, моделювання й трасування схем та РСВ);

FGSV – Forschungsgesellschaft für Straßen – und Verkehrswesen e. V. (німецьке технічно-наукове товариство, яке розробляє та оновлює технічні правила для дорожнього та транспортного господарства);

Gerber – стандартний файловий формат для опису шарів РСВ (трасування, паяльна маска, шелкографія), необхідний для передачі даних у виробництво;

HSE – High-Speed External (зовнішнє високочастотне джерело тактування для мікроконтролера);

I2C – послідовна шина даних для зв'язку інтегральних схем;

IDE – Integrated Development Environment;

IDE – середовище розробки;

MCU – Microcontroller Unit;

PCB – Printed Circuit Board) (друкована плата, на якій розміщуються електронні компоненти і здійснюються їхні електричні з'єднання за допомогою мідних доріжок та отворів);

RiLSA – Richtlinien für Lichtsignalanlagen (рекомендації з організації та управління світлофорними системами);

RTC – Real-Time Clock (годинник реального часу).

## ВСТУП

Розробка автоматизованої системи контролю міського трафіку на базі мікроконтролера STM32 має високий рівень актуальності в умовах сучасного розвитку міської інфраструктури. Системи управління світлофорами та контролю транспортного потоку є важливою ланкою в забезпеченні безпеки на дорогах та оптимізації міського трафіку, особливо з огляду на зростаючу кількість автомобілів і вимоги до ефективності дорожнього руху.

Дослідження різноманітних підходів до автоматизації управління світлофорами, аналіз їхніх переваг і недоліків, а також вивчення сучасних технологій управління трафіком є ключовими аспектами для забезпечення стабільної та ефективної роботи системи. Розуміння особливостей застосування мікроконтролерних систем, таких як STM32, дозволяє створити надійне рішення, яке підвищує ефективність роботи світлофорів і забезпечує безпеку дорожнього руху [1].

Створення такої системи для комерційного використання особливо актуальне, адже вона сприяє оптимізації транспортного потоку, зменшенню затримок на перехрестях та підвищенню загального рівня безпеки. Розроблена система базується на використанні недорогих, але високонадійних компонентів, які можна оперативно замінити у разі виходу з ладу, що гарантує її адаптивність і довговічність.

Таким чином, метою роботи є розробка макету друкованої плати яка буде сумісна для актуальних приладів керування трафіком на базі мікроконтролера STM32 для підвищення ефективності управління світлофорами та оптимізації транспортного потоку.

Мета роботи – підвищення ефективності управління міським трафіком за допомогою автоматизації світлофорів на базі мікроконтролера STM32.

Об'єкт розробки – процес автоматизації управління світлофорами в умовах міського трафіку.

Предмет розробки – макет автоматизованої системи контролю міського трафіку на базі мікроконтролера STM32.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- аналіз німецьких стандартів DIN VDE;
- аналіз автоматизації приладу керування світлофором;
- вибір мікроконтролера;
- тестування функцій мікроконтролера;
- розробка друкованої плати;
- інтеграція друкованої плати в комплекс автоматизації.

Кваліфікаційна робота виконана згідно [2-4]. Проведені дослідження відповідають цілям сталого розвитку (ЦСР) 9 так як наукові дослідження та інновації є важливими факторами, що сприяють сталому розвитку.

# 1 АНАЛІЗ СТАНДАРТІВ ТА ВИМОГ ДО СИСТЕМИ КЕРУВАННЯ

## 1.1 Аналіз стандартів DIN VDE

Німецькі стандарти DIN та VDE мають довгу історію та відіграють ключову роль у забезпеченні якості, безпеки та ефективності виробів як у Німеччині, так і в усьому світі [5].

DIN (Deutsches Institut für Normung) було засновано в Німеччині для створення єдиних стандартів, що допомагають забезпечити сумісність і високу якість продукції. Спочатку ці стандарти розроблялися для відновлення виробництва після воєнних руйнувань, а згодом DIN став основою для гармонізації технічних вимог у багатьох галузях, таких як машинобудування, будівництво, електротехніка та інформаційні технології. Сучасні стандарти DIN є важливими елементами європейської системи стандартизації, що сприяє спрощенню міжнародної торгівлі та інтеграції на глобальному ринку.

VDE (Verband der Elektrotechnik, Elektronik und Informationstechnik) зосереджений на електротехнічних нормах. Початково стандарти VDE розроблялися для регулювання безпечної експлуатації електричних пристроїв у Німеччині. Завдяки строгим вимогам до тестування та контролю якості, норми VDE стали зразком для інших країн і інтегровані в загальноєвропейську систему стандартів. Вони визначають правила монтажу, експлуатації та перевірки електрообладнання, що гарантує надійність і безпеку систем, особливо у сфері автоматизації [6, 7].

В цьому проєкті з розробки автоматизованої системи контролю міського трафіку ці стандарти виступають орієнтирами при виборі компонентів, мови програмування та розробці технологічних процесів. Завдяки DIN і VDE було описано подальший процес автоматизації

світлофорів таким чином, щоб забезпечити сумісність, безпеку та стабільність роботи всієї системи. Дотримання цих норм дозволяє впевнено інтегрувати розроблене рішення в існуючі інфраструктури, а також відповідає вимогам сучасного ринку як в Європі, так і у світовій практиці.

### 1.1.1 DIN V VDE V 0823-500

Стандарт DIN V VDE V 0823-500 є частиною німецьких нормативних документів, розроблених для регулювання технічних вимог та процедур безпеки для електронних систем управління дорожньо-транспортним рухом, зокрема – світлофорних систем. Він інтегрує вимоги як з боку німецького інституту стандартизації (DIN), так і з боку німецької асоціації електротехніки (VDE), спрямовані на забезпечення стабільності, надійності, безпеки та ефективності роботи апаратно-програмних комплексів, що використовуються у дорожньому регулюванні [7].

Основні цілі стандарту:

- підвищення безпеки дорожнього руху: стандарт встановлює критерії для проектування та експлуатації систем, які гарантують, що у випадках аварійних ситуацій або відмов окремих компонентів система автоматично переходить у режим, що мінімізує ризики для учасників дорожнього руху;

- забезпечення високої надійності та стабільності роботи: застосування чітко визначених алгоритмів перемикання сигналів, контрольних інтервалів та режимів самодіагностики дозволяє досягти високої надійності систем навіть у суворих умовах експлуатації;

- гармонізація з іншими міжнародними та європейськими стандартами: використання цього стандарту сприяє інтеграції розроблених систем у єдиний стандартний комплекс, сумісний із вимогами ISO, IEC, EN та іншими нормами, що полегшує інтеграцію систем керування в міжнародну інфраструктуру.

Ключові вимоги стандарту:

- стандарт визначає допустимі межі для часових інтервалів світлових

сигналів (зелений, жовтий, червоний), враховуючи фактори, такі як час затухання (clearance interval) та середній час реакції водіїв. Наприклад, необхідно забезпечити достатній час для повного розчищення перехрестя перед початком наступної фази;

- система самодіагностики та безпеки: вбудовані механізми самоперевірки та автоматичного переходу в аварійний режим є обов'язковими. У разі виявлення відмови ключових сигналів (наприклад, червоного сигналу для транспортних засобів), система повинна відразу впровадити алгоритми аварійного реагування або вимкнення для запобігання небезпеці;

- інтеграція з апаратною та програмною частинами: стандарт регламентує взаємодію мікроконтролерів, датчиків та інших апаратних компонентів з програмною логікою системи. Це включає реалізацію алгоритмів керування, обробки сигналів у режимі реального часу та можливість адаптивного реагування на змінні умови дорожнього руху (наприклад, через зміну інтенсивності потоку);

- електромагнітна сумісність (ЕМС) та захищеність: з метою забезпечення стійкості роботи в умовах впливу електромагнітних перешкод стандарт задає вимоги до захищеності електронних компонентів і систем зв'язку, що є важливим аспектом для запобігання несправностей через зовнішній вплив;

- вимоги до документації та контролю якості: стандарт накладає обов'язок вести повну технічну документацію, що включає протоколи тестування, результати калібрування та звіти щодо експлуатації. Це сприяє відстеженню продуктивності системи протягом її життєвого циклу та можливості своєї модернізації.

## 1.2 Аналіз правил RiLSA

RiLSA (Guidelines for Traffic Signals) – це комплекс рекомендацій та правил, розроблених німецькою організацією FGSV (Forschungsgesellschaft für Straßen- und Verkehrswesen) для організації та управління світлофорними системами. Цей документ має довгу історію, адже вперше його принципи були сформульовані для підвищення безпеки дорожнього руху та оптимізації транспортного потоку в Німеччині, а згодом стали орієнтиром для світового співтовариства [8-9].

### 1.2.1 Основні аспекти RiLSA

Основні принципи та методи проектування: RiLSA описує фундаментальні підходи до організації світлофорних систем. Документ визначає, як правильно розташовувати сигнали, встановлювати оптимальні режими роботи та забезпечувати максимальну видимість для всіх учасників дорожнього руху:

- зв'язок із дорожньою інфраструктурою: одним із важливих аспектів є інтеграція світлофорних систем з конструкцією та розташуванням дорожніх споруд. RiLSA підкреслює, що правильне поєднання технічного проектування доріг із принципами роботи сигналізації сприяє безпечному та ефективному руху транспорту;

- контрольні методи та алгоритми управління: документ містить рекомендації щодо використання сучасних технологій, включаючи автоматизацію, застосування датчиків руху та розробку алгоритмів керування, що дозволяють адаптувати роботу світлофорів до змінних умов дорожнього руху;

- технічний дизайн та вимоги до якості: RiLSA містить розділи, присвячені технічному дизайну світлофорних систем, їх технічному приймальному контролю та експлуатації. Рекомендації спрямовані на забезпечення високої надійності та безпеки систем, що відповідають як

національним, так і міжнародним стандартам;

– управління якістю та обслуговування: важливим елементом є система контролю якості, яка гарантує, що встановлені системи відповідають сучасним вимогам безпеки та ефективності. Це включає регулярне тестування, перевірку компонентів та аналіз ефективності роботи системи.

### 1.2.2 Історичний контекст та міжнародна інтеграція

RiLSA має глибокі історичні корені. Вперше принципи регулювання світлофорних систем були розроблені в Німеччині, де зростання автомобільного транспорту вимагало створення ефективних рішень для управління дорожнім рухом. З часом ці принципи були адаптовані та модернізовані, щоб відповідати сучасним технологічним вимогам [8].

Сучасна редакція RiLSA (видання 2015 року з перекладом 2021 року) відображає останні досягнення в галузі дорожнього управління та автоматизації. Вона інтегрує рекомендації, що відповідають міжнародним стандартам, зокрема нормам ISO та IEC, а також гармонізується з європейськими стандартами (EN, CEN, CENELEC). Це забезпечує сумісність та легкість інтеграції світлофорних систем у різних країнах, сприяючи спрощенню міжнародної торгівлі та обміну технологіями.

В розробці автоматизованої системи контролю міського трафіку [10] основні принципи RiLSA слугують орієнтиром для проектування системи управління світлофорами. Завдяки використанню рекомендацій цього стандарту може бути забезпечено:

– високу надійність системи: використання перевірених методів проектування та управління гарантує стабільну роботу компонентів навіть у складних умовах;

– безпеку дорожнього руху: оптимізація режимів роботи світлофорів сприяє зниженню аварійності та покращенню комунікації між різними учасниками дорожнього руху;

– сумісність із сучасними технологіями: гармонізація з міжнародними

та європейськими стандартами дозволяє інтегрувати розроблене рішення у вже існуючу інфраструктуру, що відповідає сучасним вимогам ринку.

Таким чином, RiLSA не тільки визначає технічні вимоги до світлофорних систем, але й є фундаментом для розробки ефективних, безпечних та сучасних технологій управління міським трафіком, що мають значення як на національному, так і на міжнародному рівнях.

### 1.3 Аналіз вимог до створення приладу автоматизованого керування

Автоматизація приладу керування світлофорами, що базується на вимогах стандарту DIN V VDE V 0823-500 та рекомендаціях RiLSA (Guidelines for Traffic Signal Systems), потребує системного підходу, який охоплює всі етапи – від аналізу нормативної бази до розробки архітектури програмного забезпечення, реалізації алгоритмів керування та інтеграції з апаратною частиною. Система повинна відповідати високим вимогам безпеки, надійності та адаптивності, що забезпечує оптимізацію дорожнього руху у сучасному міському середовищі [11].

#### 1.3.1 Основні аспекти автоматизації приладу керування світлофорами:

##### а) геометрія та розміщення:

– розташування світлофорів та допоміжних дорожніх маркувань має відповідати нормативним вимогам для забезпечення безпечного руху пішоходів та транспортних засобів;

– відстань утримувальних ліній від пішохідного переходу визначається з метою запобігання конфліктів між учасниками дорожнього руху;

##### б) розрахунок часів сигналів:

– зелений сигнал для пішоходів: тривалість зеленого світла розраховується з урахуванням середньої швидкості пішохода та розміру пішохідного переходу;

– червоний сигнал для транспортних засобів: час червоного сигналу визначається з урахуванням часу, необхідного для повного розчищення перехрестя, а також часу реакції водіїв;

в) фазова логіка перемикавання:

– автоматизація повинна забезпечувати чітко та послідовне перемикавання між різними фазами (зелений, жовтий, червоний) з урахуванням змінної інтенсивності руху (рис. 1.1);

– обов'язковим є «час затухання» (clearance interval), який забезпечує повне розчищення перехрестя перед початком наступної фази.

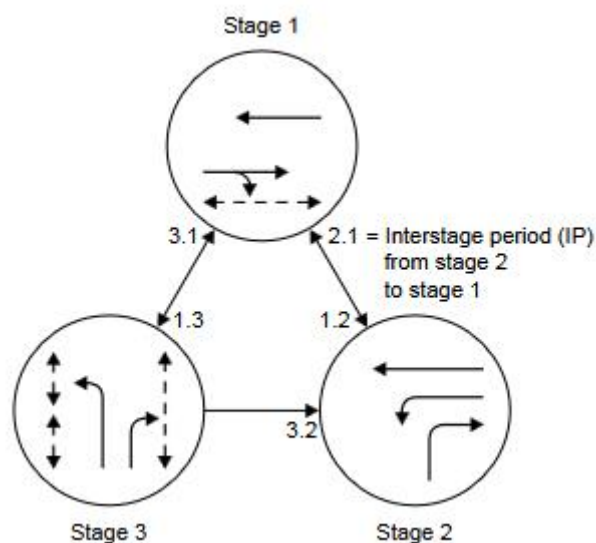


Рисунок 1.1 – Приклад діаграми послідовності етапів [11]

г) системи безпеки та відмовостійкості:

– прилад керування повинен бути обладнаний механізмами самодіагностики, що постійно контролюють працездатність компонентів системи;

– у разі виявлення несправності (наприклад, відсутності червоного сигналу для транспортних засобів) система має автоматично перейти в аварійний режим або вимкнутися для запобігання аварійним ситуаціям;

д) інтеграція з іншими системами управління:

– прилад керування повинен бути сумісним із системами моніторингу та регулювання транспортного потоку міста, що дозволяє

адаптувати режим роботи в режимі реального часу;

е) логіка безпеки:

– у разі виявлення збоїв, наприклад, якщо червоний сигнал для транспортних засобів неактивний, програмне забезпечення має автоматично перевести систему в аварійний режим;

– реалізація цього механізму базується на умовній логіці: якщо (сигнал червоний для транспорту неактивний) – активувати аварійний режим (автоматичне вимкнення або переходу в режим «безпеки») [12].

Основні формули, що застосовуються в розрахунках:

– час зеленого сигналу для пішоходів:

$$T_{\text{green\_ped}_j} = \frac{(L_{\text{cross}_j} + S_{\text{margin}_j})}{v_{\text{avg}_j}} \quad (1.1)$$

де  $L_{\text{cross}_j}$  – довжина пішохідного переходу;

$S_{\text{margin}_j}$  – запас безпеки (додаткова відстань для непередбачуваних затримок);

$v_{\text{avg}_j}$  – середня швидкість пішохода;

– час червоного сигналу для транспортних засобів:

$$T_{\text{red\_vehicle}_j} = T_{\text{clearance}_j} + t_{\text{reaction}_j} \quad (1.2)$$

де  $T_{\text{clearance}_j}$  – час, необхідний для повного розчищення перехрестя;

$t_{\text{reaction}_j}$  – середній час реакції водія;

– час затухання (clearance interval):

$$T_{\text{clearance}_j} = \frac{d}{v_{\text{vehicle}_j}} \quad (1.3)$$

де  $d$  – відстань, яку має проїхати транспортний засіб для повного виїзду з перехрестя;

$v_{(vehicle)}$  – середня швидкість транспортного засобу під час розчищення;

– загальний цикл роботи світлофора:

$$T_{(cycle)} = T_{(green)} + T_{(yellow)} + T_{(red)} \quad (1.4)$$

де  $T_{(green)}$  – час зеленого сигналу;

$T_{(yellow)}$  – час жовтого сигналу;

$T_{(red)}$  – час червоного сигналу;

– адаптивне керування режимом зеленого сигналу:

$$T_{(green\_adaptive)} = T_{(green\_nominal)} \times k_{(load)} \quad (1.5)$$

де  $T_{(green\_nominal)}$  – номінальний час зеленого сигналу;

$k_{(load)}$  – коефіцієнт навантаження, що відображає поточну інтенсивність руху (значення більше 1 – збільшення часу зеленого, менше 1 – зменшення).

#### 1.4 Допоміжні інструменти та пристрої для автоматизації та контролю трафіку

Сучасні системи автоматизації дорожнього руху спрямовані на оптимізацію транспортного потоку, підвищення рівня безпеки та інтеграцію інноваційних технологій у міську інфраструктуру. Для досягнення цих цілей використовуються різні допоміжні прилади, які забезпечують збір даних, взаємодію користувачів із системою керування та адаптивне регулювання світлофорних режимів. Відповідно до нормативних документів DIN V VDE V 0823-500 та рекомендацій RiLSA, ці прилади повинні відповідати високим стандартам надійності, безпеки та ефективності [13].

### 1.4.1 Пішохідні кнопки виклику сигналу

Пішохідні кнопки є ключовим елементом взаємодії між пішоходами та системою керування світлофорами. В Німеччині дуже популярні рішення від компанії RTB – вони використовуються як для класичних тактильних кнопок, так і для пристроїв, оснащених додатковою акустичною та вібро-функцією, основні з них:

– класичні тактильні кнопки від RTB: ці пристрої призначені для стандартного виклику пішохідного сигналу. При натисканні кнопки активується електронний сигнал, який передається до центрального блоку керування. Індикація може здійснюватися за допомогою світлових індикаторів (LED), що забезпечує зрозуміле повідомлення користувачу про успішну реєстрацію запиту. Корпуси кнопок виготовляються із зносостійких матеріалів, що відповідають вимогам експлуатації на відкритих майданчиках (рис. 1.2);



Рисунок 1.2 – Класична кнопка виклику сигналу від RTB

– кнопки з вібро-функцією для осіб із порушеннями зору від RTB: ці моделі поєднують тактильний відгук з вібрацією, що дозволяє користувачам з порушеннями зору отримати надійне підтвердження активації сигналу навіть у відсутності зорових індикаторів. Деякі рішення також інтегрують акустичну індикацію, яка голосово повідомляє про реєстрацію запиту (наприклад, фраза «Сигнал виклику зареєстровано»). Поєднання цих

технологій від RTB робить систему максимально інклюзивною та зручною для всіх категорій користувачів.

#### 1.4.2 Індукційні петлі

Індукційні петлі (рис. 1.3) є невід'ємною складовою систем збору даних у системах управління дорожнім рухом. Їх встановлюють безпосередньо у дорожньому покритті на ділянках, де необхідно визначати наявність транспортних засобів [12-13].



Рисунок 1.3 – Зовнішній вигляд індукційної петлі

1.4.2.1 Принцип роботи: основою роботи індукційних петель є формування електромагнітного поля. Коли транспортний засіб з великою кількістю металу проїжджає над петлею, відбувається зміна цього поля, що фіксується спеціальними сенсорами.

Ці дані потім використовуються для визначення часу прибуття та кількості транспортних засобів, що дозволяє адаптивно коригувати час перемикання світлофорів.

1.4.2.2 Приклади рішень: популярні системи від виробників, таких як Kapsch, TrafficCom або SWARCO, пропонують моделі індукційних петель, що забезпечують високу точність та швидку обробку даних у режимі реального часу. Таке обладнання дозволяє ефективно координувати роботу

світлофорів на перехрестях, враховуючи змінну інтенсивність руху.

Пристрій забезпечує рівномірне та контрольоване опромінення, що дозволяє досягти високої міцності та стабільності деталей (рис. 1.3).

Регульовані параметри опромінення та контроль температури забезпечують оптимальні умови для різних типів смол та вимог проекту.

#### 1.4.3 Камери відеоспостереження

Системи відеоспостереження відіграють важливу роль у моніторингу та аналізі дорожнього руху. Сучасні відеокамери забезпечують як високоякісну відеозйомку, так і можливість автоматичного аналізу отриманих зображень.

##### 1.4.3.1 Основні характеристики

Камери дозволяють відстежувати транспортний потік, здійснювати класифікацію транспортних засобів і виявляти аварійні ситуації. Завдяки інтеграції з алгоритмами розпізнавання образів, камери можуть автоматично аналізувати дані для подальшого коригування світлофорних режимів.

##### 1.4.3.2 Актуальні ринкові рішення

У Німеччині новітні системи відеоспостереження базуються на технологіях від компанії FLIR. Наприклад, моделі, які використовують тепловізійні та високочутливі сенсори FLIR (рис. 1.4), забезпечують відмінну якість зображення навіть при низькому освітленні або несприятливих погодних умовах.

Такі камери дозволяють оперативно реагувати на змінні сценарії дорожнього руху і підвищувати загальний рівень безпеки на перехрестях (рис. 1.5).



Рисунок 1.4 – Інфрачервона камера від FLIR  
(модель : ThermiCam AI – 632)



а)

б)

Рисунок 1.5 – Інфрачервоне (теплове) зображення міської дороги з автоматичним виявленням транспортних засобів за допомогою інфрачервоної камери FLIR [11]

На рис. 1.5, а автомобілі позначені прямокутниками з вказаними ймовірностями розпізнавання та розрахованими швидкостями (км/год), на рис. 1.5, б – перспективна топ-даун схема розташування цих транспортних засобів у смугах руху.

Допоміжні пристрої інтегруються в єдину систему автоматизації дорожнього руху, де вони виконують такі основні функції:

- оперативний збір даних: індукційні петлі, камери та пішохідні кнопки забезпечують постійний моніторинг дорожньої ситуації, що дозволяє центральній системі приймати швидкі та точні рішення;

- оперативна взаємодія: пішохідні кнопки з вібро- та акустичними відгуками (від RTB) забезпечують користувачам миттєве підтвердження активації сигналу, незалежно від їх зорових можливостей;

- адаптивне управління: зібрані дані інтегруються в центральний блок керування, який аналізує інтенсивність руху та поточні умови, коригуючи час перемикання світлофорних режимів для оптимізації транспортного потоку та зниження аварійності;

- сумісність із сучасною інфраструктурою: використання бездротових технологій та систем V2X забезпечує можливість інтеграції допоміжних пристроїв у загальну систему управління міським рухом, що дозволяє оперативно реагувати на змінні умови і аварійні ситуації.

## 2 ВИБІР ТА НАЛАШТУВАННЯ МІКРОКОНТРОЛЕРА

### 2.1 Вибір мікроконтролера

Під час другого етапу буде проведено аналіз обраного мікроконтролера, обґрунтування його вибору а вибір підходящої IDE для подальшої розробки та програмування.

Для реалізації апаратної частини було обрано промислову плату DevEBox (рис. 2.1) STM32H7XX\_M, яка випускається в двох варіантах модулів – з процесором STM32H743VIT6 та більш бюджетним STM32H750VBT6.

DevEBox вирізняється компактним форм-фактором (40×68 мм) та доступністю пін-совісної розводки з іншими модулями серії STM32.

Ця платформа містить весь необхідний набір периферії (USB-DFU для оновлення прошивки, Ethernet, CAN, UART, SPI, I<sup>2</sup>C), що дозволяє мінімізувати додаткові перехідні плати й підвищити надійність системи.

Також плата DevEBox STM32H7XX\_M оснащена слотом microSD, підключеним до вбудованого контролера SDMMC. Він підтримує формати SD, SDIO та MMC у режимах DS, HS та SDR (1-, 4- та 8-бітні шини), а також працює з внутрішнім DMA для мінімізації навантаження на ядро. Це дає змогу зберігати великі логи роботи системи, архіви камер та конфігураційні файли без додаткових зовнішніх модулів.

Максимальна швидкість передачі – до 208 МБ/с залежно від карти та налаштувань шини.

Сумісність – повна підтримка SD 4.1 та MMC 4.51; backward compatibility із попередніми версіями карт.

Застосування – архівація відеопотоків із камер FLIR, зберігання історії детекції індукційних петель, оновлення конфігурації “на льоту”.

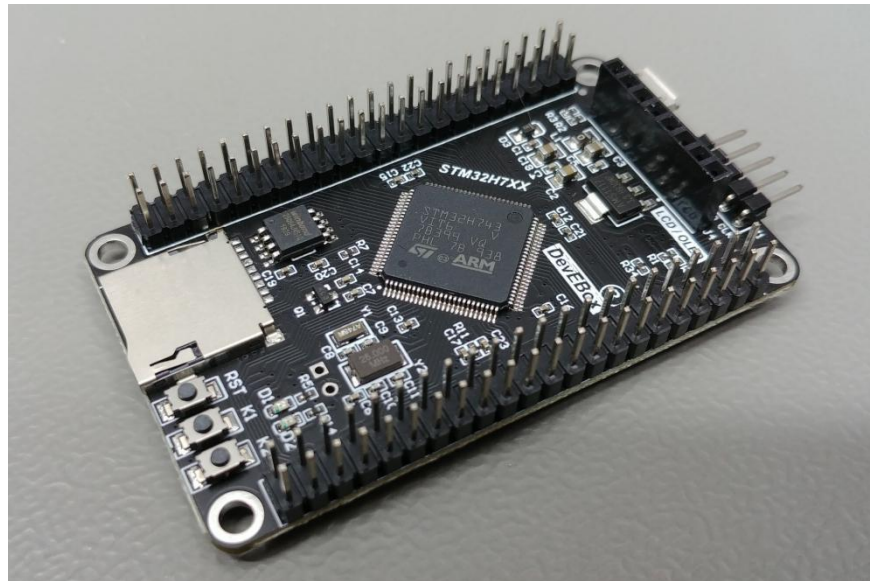


Рисунок 2.1 – Зовнішній вигляд китайського мікроконтролеру DevEBox на базі процесору STM32H743VIT6

### 2.1.1 Детальний опис процесора STM32H743VIT6

Процесор STM32H743VIT6 побудований на основі високопродуктивного ядра ARM Cortex-M7 із тактовою частотою до 480 МГц. Завдяки вбудованому FPU (Floating-Point Unit) і розширеним DSP-інструкціям він здатен обробляти складні математичні операції – наприклад, фільтрацію сигналів з індукційних петель і розрахунок фазової логіки світлофорів – із мінімальними затримками.

Пристрій має два рівні кеш-пам'яті L1: інструкційний (I-cache) та даних (D-cache). Це дозволяє знизити латентність доступу до внутрішньої Flash-пам'яті і підвищити загальну пропускну здатність шини команд, що критично для систем із жорсткими вимогами до реального часу.

STM32H743VIT6 оснащений 2 МБ Flash-пам'яті, організованої в двобанкову архітектуру, а також 1 МБ швидкої SRAM, частина якої може бути виділена під DTCM (Data-Tightly Coupled Memory) для критичних по швидкодії фрагментів коду. Така конфігурація дозволяє розмістити одночасно і самі алгоритми керування, і буфери даних від сенсорів, та звести до мінімуму затримки на доступ до пам'яті.

Апаратно-програмна платформа має три канали DMA, кожен з яких

може обслуговувати до 16 потоків передачі даних між пам'яттю та периферією без участі ядра. Це звільняє процесор для виконання обчислювальних завдань, залишаючи передачу кадрів з камер і зчитування сигналів з індукційних петель на автоматичний режим.

Система тактування STM32H743VIT6 побудована на гнучкій PLL-мережі: окремі PLL дозволяють генерувати необхідні високі тактові частоти для ядра, периферійних шин і зовнішніх інтерфейсів (ETH, USB, SDMMC). Для кожного з шин AHB1, AHB2, APB1, APB2 передбачені незалежні прескалери, що надає можливість тонко налаштувати енергоспоживання та швидкодію окремих блоків.

У розпорядженні проектувальника – більш ніж двадцять таймерів різних типів (Basic, General-purpose, Advanced), серед яких є й ті, що підтримують режим PWM із високою роздільною здатністю. Це дає змогу реалізувати точне керування інтенсивністю світла світлофорів та сигнальними індикаторами.

З безпекових механізмів у мікроконтролері вбудовано MPU (Memory Protection Unit) для ізоляції критичних ділянок коду, PC-ROP (Program Counter-Read-Only Protection) для блокування запису в область Flash під час виконання, а також апаратний модуль Active Tamper для виявлення спроб несанкціонованого доступу.

STM32H743VIT6 підтримує безліч популярних інтерфейсів:

- Ethernet MAC із підтримкою RMIІ та MII, що дозволяє безпосередньо під'єднувати контролер до мережі міського управління;
- CAN-FD для високошвидкісної комунікації із суміжними вузлами інфраструктури;
- USB OTG HS із зовнішнім PHY, що використовується для прошивки пристрою в режимі DFU;
- SPI/I<sup>2</sup>C/UART для підключення датчиків і виконавчих пристроїв;
- SDMMC для читання/запису логів на картку пам'яті;
- ADC та DAC для аналого-цифрового та цифро-аналогового

перетворення сигналів від фоторезисторів і світлодіодних індикаторів.

Також були розглянуті інші претенденти на роль центрального контролера, але вони не підходять через низку причин:

а) Raspberry Pi

1) потребує значно більших апаратних ресурсів (живлення 5 V/3 A, охолодження для роботи до 10 W), що ускладнює інтеграцію в систему з жорсткими вимогами до реального часу;

2) вища ціна – Raspberry Pi 4 Model B (4 GB) коштує близько 35 USD, що перевищує вартість Arduino Uno Rev3  $\approx$  27,60 USD;

б) Arduino Uno (ATmega328P)

1) обмежена продуктивність (16 MHz, 2 KB SRAM, 32 KB Flash) недостатня для обробки відеоданих і складних адаптивних алгоритмів фазового керування;

в) ESP32

1) незважаючи на подвійне ядро і Wi-Fi/Bluetooth, архітектура складніша для гарантування детермінованості real-time завдань, а ADC мають нижчу лінійність, що ускладнює точне зчитування аналогових сигналів від сенсорів.

Таким чином обраний мікроконтролер STM32H743VIT6 у складі плати DevEBox STM32H7XX\_M є ідеальним рішенням для автоматизованої системи керування світлофорами завдяки поєднанню високої обчислювальної потужності (ARM Cortex-M7 @ 480 МГц, FPU, DSP) з об'ємом пам'яті 2 МБ Flash і 1 МБ SRAM, що забезпечує своєчасну обробку складних алгоритмів фазового керування. Широкий набір периферійних інтерфейсів (Ethernet, CAN-FD, USB, SPI, I<sup>2</sup>C, ADC/DAC) дозволяє безпосередньо інтегрувати індукційні петлі, тепловізійні камери і пішохідні кнопки.

Підтримка FreeRTOS/CMSIS-RTX через екосистему STM32Cube спрощує розробку детермінованих багатозадачних застосунків, а вбудована підтримка слота microSD забезпечує надійне зберігання логів і відеоархівів без зовнішніх пристроїв. Низьке енергоспоживання, багатоступеневі режими

Power-Save та конкурентна ціна (~16 USD за MCU) роблять його економічно вигідним і технологічно виправданим вибором для масштабних впроваджень.

## 2.2 Вибір програмного фреймворку

У ході аналізу було розглянуто кілька вбудованих фреймворків, сумісних із мікроконтролером STM32H743VIT6 на платі DevEBox STM32H7XX\_M: рідний STM32Cube, FreeRTOS, Zephyr, Mbed OS, ChibiOS та PlatformIO STM32Cube. Порівняння враховувало критерії інтеграції, портативності, підтримки middleware, простоти налаштування та ліцензії. У результаті рідний фреймворк STM32Cube виявився оптимальним вибором завдяки тісній інтеграції з апаратною платформою, великій кількості готових middleware-компонентів, відповідності стандартам якості та безпеки і безкоштовній BSD-ліцензії.

### 2.2.1 STM32Cube (HAL/LL + middleware)

STM32CubeH7 – це повноцінний пакет від STMicroelectronics, що включає HAL-LL драйвери, CMSIS-Core, BSP для плат DevEBox і middleware (RTOS, USB, FATFS, TCP/IP, графіка) STMicroelectronics.

- HAL-LL абстрагування дає єдиний інтерфейс для всіх серій STM32 із високою портативністю коду GitHub;
- повний набір прикладів для STM32H7 прискорює старт розробки та тестування STMicroelectronics;
- інтеграція з STM32CubeMX дозволяє генерувати налаштування периферії в графічному середовищі Gab;
- BSD-ліцензія забезпечує гнучкість комерційного використання без додаткових витрат GitHub.

### 2.2.2 FreeRTOS/CMSIS-RTOS RTX

FreeRTOS – найпоширеніша RTOS з підтримкою STM32Cube

middleware. STM32CubeH7 містить порт FreeRTOS з готовими налаштуваннями для диспетчеризації завдань, таймерів, черг і семафорів PlatformIO Registry:

- висока детермінованість при пріоритетному виконанні завдань і підтримка сервісів TCP/IP stack, USB, FATFS Mouser;
- легка інтеграція в STM32CubeMX/IDE, але вимушує розробника працювати з окремими .ioc-файлами і middleware-компонентами.

### 2.2.3 Zephyr RTOS

Zephyr – модульна, безпечна та масштабована RTOS із підтримкою STM32H7 через офіційний порт у репозиторії Zephyr Project GitHub:

- підтримує мультикористувацький kernel, device tree та багато драйверів для CAN, Ethernet, SPI, I<sup>2</sup>C community.st.com;
- вимагає складнішого вивчення device tree та west build system, порівняно з CubeMX;
- активна спільнота і регулярні релізи, але менше готових демонстрацій для специфічних плат ST.

### 2.2.4 Mbed OS

Mbed OS надає зручний онлайн-компілятор і готові бібліотеки для NUCLEO-H743ZI та DISCO-H747I os.mbed.com:

- абстрагування мережі та файлової системи, але підтримка Ethernet на STM32H7 наразі обмежена os.mbed.com;
- Порт для STM32H7 вимагає додаткового налаштування та іноді страждає через неповну підтримку апаратних модулів GitHub.

### 2.2.5 ChibiOS/RT

ChibiOS – компактна RTOS із низькою затримкою (RTC, NIL), оптимізована для STM32H7 на рівні HAL:

- підтримує CAN-FD, USB, SDIO, але вимагає ручного налаштування проекту (Makefile/IDE);
- добре підходить для проектів з жорсткими обмеженнями по пам'яті, але має меншу кількість сторонніх middleware.

### 2.2.6 PlatformIO STM32CubeH7

PlatformIO інтегрує STM32CubeMX код із відкритим середовищем, дозволяє працювати в VS Code чи Atom через stm32pio. Він дає крос-платформену збірку і відлагодження, але розробник зобов'язаний налаштувати версії HAL/LL вручну.

Вибір рідного фреймворку STM32Cube обґрунтовано такими перевагами:

- глибока інтеграція з апаратурою DevEBox STM32H7XX\_M та підтримка всіх інтерфейсів (USB-DFU, Ethernet, CAN-FD, microSD, SDMMC) без зовнішніх бібліотек;
- включені RTOS, файлові системи, стеки TCP/IP та демонстраційні проекти для Nucleo/Discovery плат;
- мінімальна залежність від сторонніх компонентів, гарантія сумісності з оновленнями ST і відповідність стандартам MISRA C®:2004;
- підтримка корпоративного рівня валідації та тестування драйверів за допомогою CodeSonar® і додаткових верифікаційних процесів в ST.

Таким чином, STM32Cube є оптимальним фреймворком для проекту: він забезпечує повний цикл розробки від конфігурування апаратури до відлагодження готового рішення й відповідає стандартам ДСТУ та вимогам City Traffic Control.

### 2.3 Вибір середовища розробки

У процесі підготовки другорядного програмного забезпечення для контролера DevEBox STM32H7XX\_M було розглянуто кілька IDE, що

широко застосовуються для мікроконтролерів STM32: STM32CubeIDE, Keil MDK-ARM, IAR Embedded Workbench, PlatformIO (VS Code), Eclipse + System Workbench (AC6). Критерії вибору включали інтеграцію з апаратною платформою, підтримку фреймворків (HAL/LL, FreeRTOS), можливості налагодження, простоту конфігурування та ліцензійні умови.

### 2.3.1 STM32CubeIDE

Опис: вбудована IDE від ST на основі Eclipse®/CDT з інтегрованим STM32CubeMX для графічної конфігурації периферії, генерації коду та GNU Arm Toolchain.

Переваги:

- нативна підтримка всіх мікроконтролерів STM32 та middleware (FreeRTOS, USB, TCP/IP);
- зручний графічний конфігуратор периферії та тактування, що скорочує час налаштування проекту;
- потужні можливості налагодження (SWV, перехоплення переривань, візуалізація регістрів);
- безкоштовна BSD-ліцензія для комерційного використання;

Недоліки:

- інколи велика вага Eclipse та уповільнена робота на слабких ПК;
- обмежена інтеграція з альтернативними build-системами (CMake, Meson).

### 2.3.2 Keil MDK-ARM (μVision)

Опис: комерційна IDE від Arm з власним компілятором ArmCC і відлагоджувачем ULINK.

Переваги:

- високопродуктивний компілятор ArmCC з оптимізацією size/speed;
- інтегровані сервіси тестування та сертифікації (Arm Compiler Qualification).

Недоліки:

- дорогі ліцензії; безкоштовна версія обмежена 32 КБ коду;
- відсутність інтеграції з STM32CubeMX; ручне налаштування периферій.

### 2.3.3 IAR Embedded Workbench

Опис: комерційне середовище з власним компілятором IAR C/C++ та засобами аналізу коду.

Переваги:

- дуже висока оптимізація коду, підтримка статичного аналізу (C-STAT);
- повний набір інструментів профілювання та захисту безпеки.

Недоліки:

- висока вартість ліцензії, складність інтеграції з відкритими фреймворками.

### 2.3.4 PlatformIO (VS Code)

Опис: Розширення для VS Code із підтримкою багатьох плат та фреймворків, включно з STM32Cube.

Переваги:

- легка інтеграція з CMake, Arduino, Zephyr тощо;
- крос-платформна робота у звичному середовищі VS Code.

Недоліки:

- потребує додаткового налаштування для генерації коду з STM32CubeMX;
- обмежена підтримка нативних STM-демонстрацій.

### 2.3.5 Eclipse + System Workbench (AC6)

Опис: вільна IDE від AC6 на базі Eclipse із вбудованими плагінами для STM32 .

Переваги:

- безкоштовна альтернатива з підтримкою CubeMX імпорту;
- плагін debugging для ST-LINK/J-Link.

Недоліки:

- менш інтуїтивний інтерфейс, складніша інтеграція з останніми HAL-версіями.

Найкращим рішенням визнано STM32CubeIDE, оскільки воно:

- має зручний графічний конфігуратор MCU, який скорочує час первинного налаштування;
- підтримує FreeRTOS, middleware та налагодження специфічних периферій (SWV, DAP) без додаткового налаштування;
- безкоштовне розповсюдження виключає ризики ліцензійних обмежень у комерційному проєкті.

Таким чином, STM32CubeIDE (рис. 2.2) забезпечує найкраще співвідношення простоти, функціональності та вартості, що відповідає вимогам стандартів ДСТУ та бізнес-цілям проєкту.

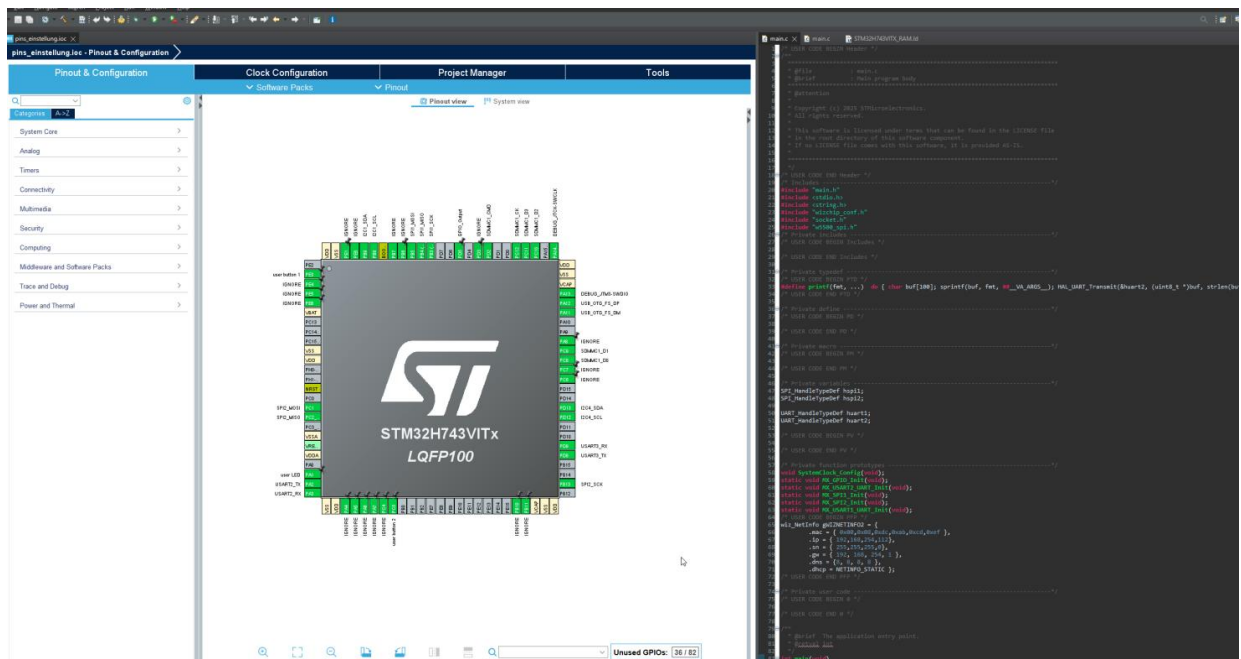


Рисунок 2.2 – Інтерфейс STM32CubeIDE

## 2.4 Налаштування функцій мікроконтролеру

Під час роботи над проектом було обрано STM32CubeIDE як основне середовище розробки. Оскільки DevEBox STM32H7XX\_M не є офіційною платою ST, всі параметри мікроконтролера необхідно прописувати вручну відповідно до рекомендацій виробника. У розділі 2.4 наведено послідовність налаштування ключових функцій STM32H743VIT6 у STM32CubeIDE для подальшої розробки друкованої плати.

### 2.4.1 Ініціалізація проекту та вибір частини

Створюємо новий проект у STM32CubeIDE, обираючи «Board Selector» → «STM32H743VIT6» (за відсутності – прописуємо вручну номер частини).

Встановлюємо назву проекту й шлях до робочої папки, обираємо GNU Arm Embedded Toolchain як компілятор (рис. 2.3).

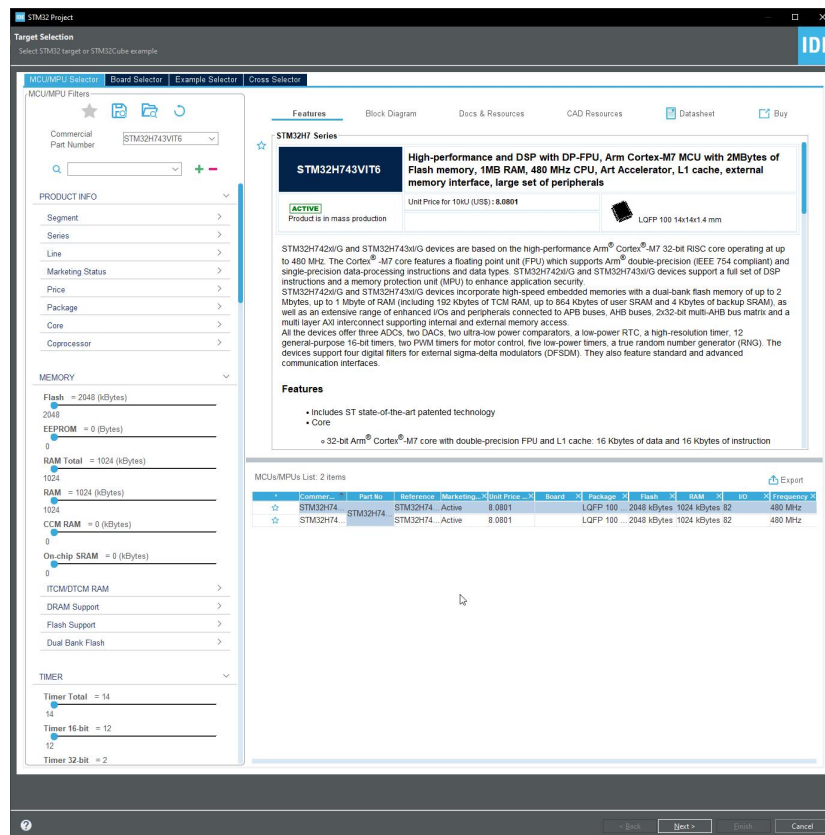


Рисунок 2.3 – Інтерфейс вибору ЦП у STM32CubeIDE

## 2.4.2 Налаштування периферії

На основі конфігурації STM32CubeIDE та потреб майбутньої плати для даного проекту були виділені такі інтерфейси: SPI1, SPI2, SDMMC1, USART2, USART3 та I2C4. Нижче описано призначення кожного з них і зв'язок із відповідними компонентами плати, яка буде пізніше побудована на їх основі.

### SPI1 (Ethernet W5500)

Модуль WIZnet W5500 (рис. 2.4) використовується як апаратний Ethernet-контролер з вбудованим TCP/IP стеком і підключається до мікроконтролера через інтерфейс SPI. У налаштуваннях CubeIDE для підключення W5500 обрано SPI1, оскільки цей канал забезпечує високошвидкісну передачу даних (W5500 підтримує SPI до 80 MHz). Таким чином, за допомогою SPI1 MCU може обмінюватися пакетами з мережею через W5500. Під час проектування друкованої плати сигнали SPI1 (SCK, MOSI, MISO, CS) буде проведено від ядра мікроконтролера до плати модуля W5500.



Рисунок 2.4 – Зовнішній вигляд модулю WIZnet W5500

### SPI2 (резерв для SD-карти)

Шина SPI2 залишена як резервний канал для можливого підключення зовнішньої SD-карти через SPI (наприклад, на додатковому адаптері). Наразі

SPI2 у проекті не задіяна, але на платі передбачено можливість використання цього порту в майбутньому. Слід зазначити, що пропускна здатність SD-карти в SPI-режимі менша, ніж через SDIO: інтерфейс SDIO (SDMMC) значно швидший. Тому за замовчуванням основним засобом роботи з SD-картою обрано SDMMC1 (див. нижче), а SPI2 лишено вільним «на випадок».

#### SDMMC1 (microSD shield)

У випадку з інтерфейсом SDMMC1 слід звернути увагу на те, що на платі DevEBox STM32H7XX\_M вже фізично реалізовано слот microSD (microSD-shield) і відповідні лінії SDMMC1 (DAT0–DAT3, CMD, CLK) виведені на конектор виробником. Тому в STM32CubeIDE ми лише активуємо контролер SDMMC1 у конфігурації периферії MCU (Pinout & Configuration → SDMMC1), не прокладаючи власних ліній під PCB-дизайн. При цьому CubeIDE генерує код налаштування SDMMC1 (ініціалізацію шин, картковий детектор, DMA), і надалі цей код можна без змін використовувати в прошивці пристрою.

Цей контролер підтримує 4-бітний режим передачі даних, що забезпечує високу швидкість обміну порівняно з SPI-режимом.

#### USART2 і USART3 (трансивер MAX3232IPW)

Порти UART (USART2 і USART3) використовуються для послідовного зв'язку з зовнішніми пристроями по стандарту RS-232. До цих портів підключено трансивери MAX3232IPW, які перетворюють TTL-лінії 3.3 В на  $\pm$ RS-232 (із зарядними насосами та рівнями до  $\pm$ 12 В). Трансивер MAX3232IPW живиться від 3.3 В (підтримує живлення 3.3–5 В) і має по два драйвери і приймачі, що дозволяє організувати по одному каналу TX/RX для кожного RS-232 порту. Наприклад, у відомому прикладі PA9/PA10 (USART1) через MAX3232 формують один порт RS-232, аналогічно в даному проекті USART2 і USART3 через два MAX3232IPW дають два незалежні RS-232 інтерфейси.

#### I2C4 (зовнішній дисплей)

Шина I2C4 призначена для підключення зовнішнього дисплея

(наприклад OLED або LCD із інтерфейсом I2C). I2C є дволінійним інтерфейсом (SDA та SCL), тому для зв'язку з дисплеєм потрібно всього дві сигнальні лінії. Наприклад, популярні OLED-дисплеї SSD1306 і LCD-модулі з I2C використовують саме такий спосіб обміну даними, що суттєво економить піни мікроконтролера.

DCMI (інтерфейс камери, відключено)

MCU STM32H743 має також спеціалізований інтерфейс камери DCMI (8–14 біт, до 80 MHz). Оскільки в цьому проекті камера не використовується, відповідні виводи DCMI в конфігурації CubeIDE встановлені в режим IGNORE. Це звільняє дані лінії для можливого іншого використання або дозволяє не трасувати їх взагалі. При необхідності в майбутньому цей інтерфейс можна активувати й підключити камеру, але наразі він виключений для спрощення схеми та уникнення непотрібних з'єднань.

#### 2.4.3 Налаштування тактування (Clock Configuration)

Для забезпечення коректної й стабільної роботи всіх периферійних модулів мікроконтролера STM32H743VIT6 необхідно виконати ретельне налаштування системи тактування. У середовищі STM32CubeIDE це завдання реалізується на вкладці Clock Configuration (рис. 2.5), де користувач одразу бачить дерево внутрішніх та зовнішніх джерел тактування, PLL-мультиплікаторів і прескелерів шин.

За основу системного годинника (SYSCLK) обрано вихідний тактовий сигнал головного фазованого синтезатора, саме він є тактовим сигналом для ядра ARM Cortex-M7 та шин AHB/APB. Далі сигнал HSI подається на D1CPRE Prescaler, який налаштовано таким чином, що на виході отримується значення 480 МГц – максимальне для ядра Cortex-M7 і периферійних шин ARM® AHB1, AHB2, APB1–4.

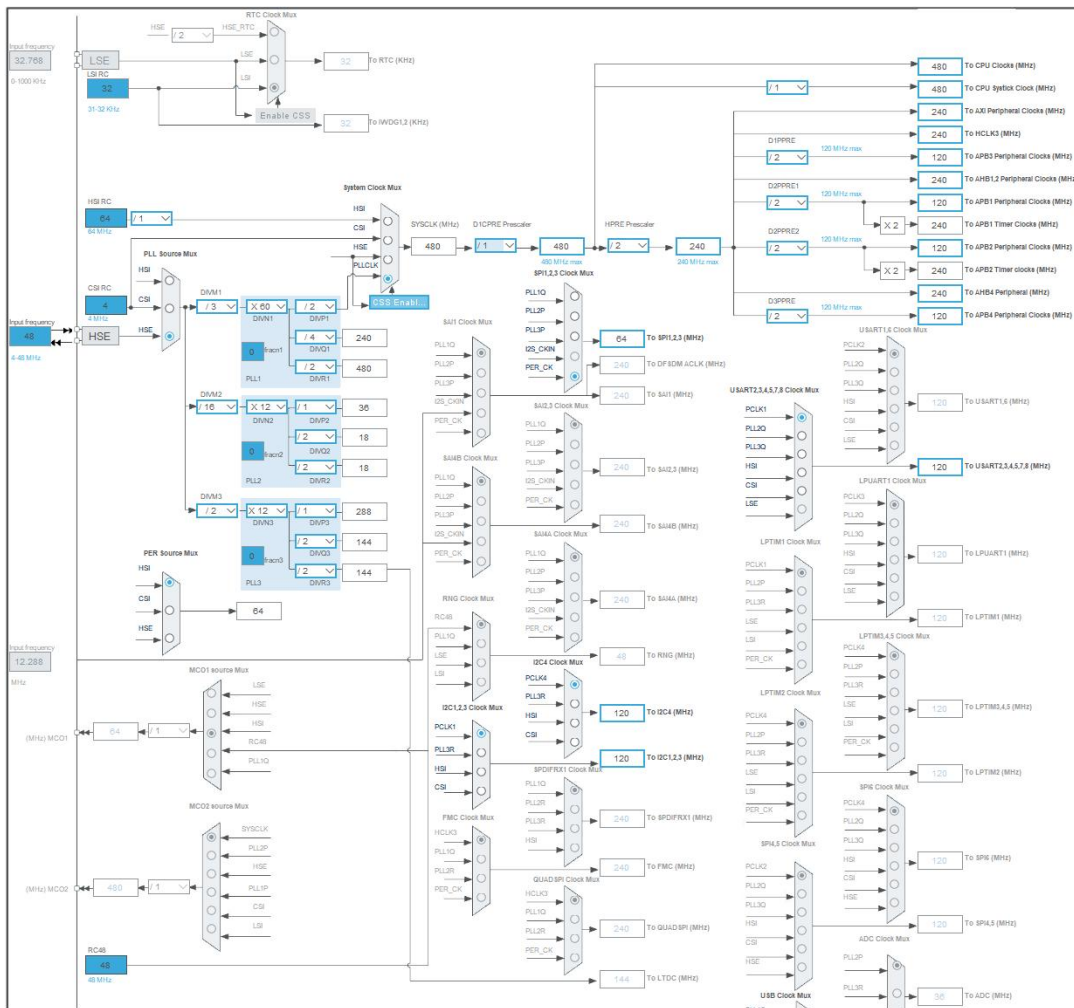


Рисунок 2.5 – Інтерфейс налаштування тактування (Clock Configuration)

Для кожного з високошвидкісних інтерфейсів, таких як SPI1, SDMMC1 та Ethernet MAC, важливо, щоби тактовий сигнал відповідав їхнім внутрішнім вимогам. Зокрема, SDMMC1 отримує свій такт від PLL1Q (divider Q), що дозволяє досягти частоти до 192 МГц для реалізації швидкісного інтерфейсу microSD у 4-бітному режимі - це критично для зберігання логів.

STM32CubeIDE автоматично перевіряє вибрані параметри на відповідність обмеженням, зазначеним у Reference Manual RM0433 і Datasheet, і попереджає про перевищення граничних частот. Після збереження конфігурації генерується код у файлі "system\_clock.c", де визначено налаштування PLL1 (PLL1M, PLL1N, PLL1P, PLL1Q), прескалери шини й ініціалізацію HSE/HSI - це дозволяє підтримувати повну

відтворюваність налаштувань при перенесенні проекту на інший комп'ютер або середовище розробки.

Налаштування тактової частоти для SPI було реалізовано за допомогою вбудованого внутрішнього RC-генератор (~64 МГц) для того щоб не було конфліктів з інтернет-модулем WIZnet W5500.

## 3 РОЗРОБЛЕННЯ ФУНКЦІОНАЛЬНОГО МАКЕТУ

### 3.1 Вибір середі проектування електроніки

У цьому підрозділі наведено детальний аналіз трьох провідних середовищ проектування електроніки (EasyEDA, Altium Designer, KiCad) з урахуванням вимог проекту автоматизованої системи керування міським трафіком. Результати показують, що EasyEDA (рис. 3.1) найкраще відповідає нашим запитам завдяки хмарній архітектурі, великій спільнотній бібліотеці, підтримці неофіційних модулів (зокрема DevEBox STM32H7XX), а також можливості швидкого замовлення прототипів плат через інтеграцію з JLCPCB.

Інші рішення (Altium Designer, KiCad) мають сильні сторони в професійному дизайні та відкритості, але вимагають значно вищих витрат часу на налаштування та обладнання, а також не забезпечують такого рівня швидкої інтеграції з виробництвом.

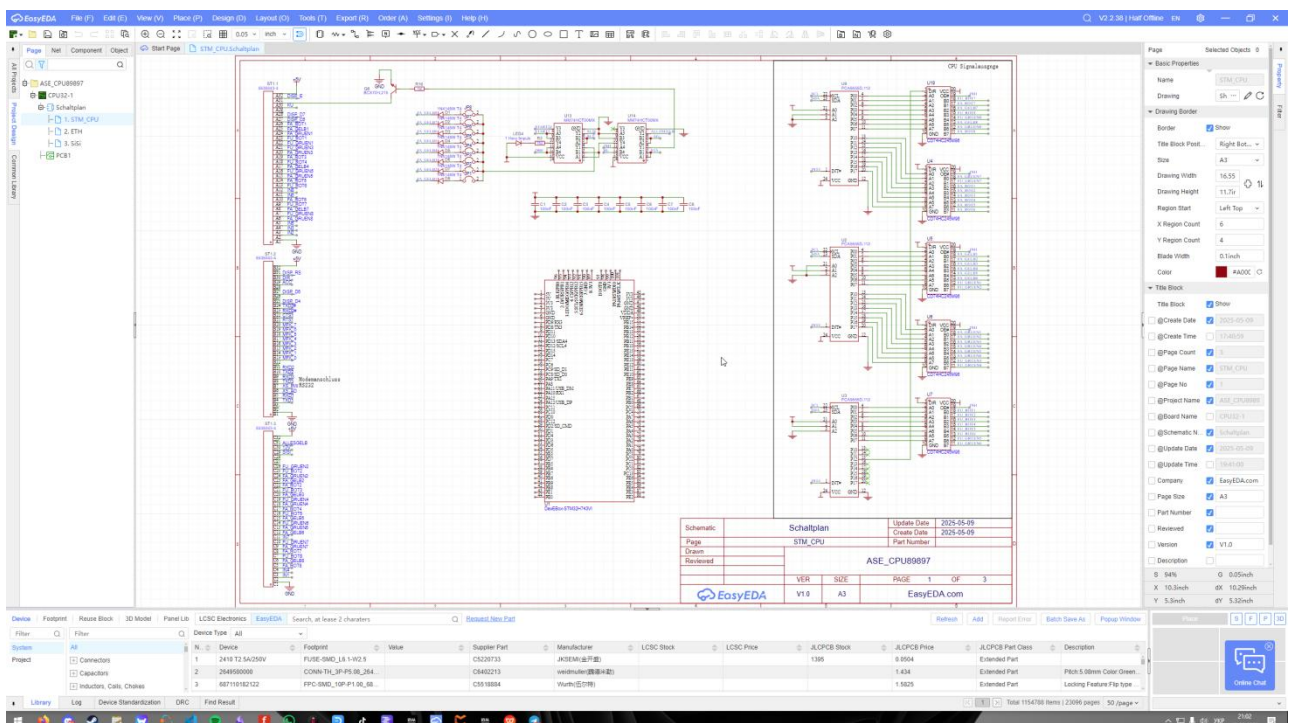


Рисунок 3.1 – Інтерфейс EasyEDA [19]

На етапі функціонального макетування було розглянуто такі інструменти:

- EasyEDA (хмарний EDA з PCB-редактором і SPICE-симуляцією);
- Altium Designer (професійна комерційна платформа для PCB-дизайну);
- KiCad (відкритий софт за ліцензією GPL для схем та PCB).

### 3.1.1 EasyEDA

EasyEDA – це веб-інструмент для розробки електроніки, що поєднує в собі схематичний редактор, SPICE-симуляцію та PCB-редактор із 2D/3D-візуалізацією і безшовною інтеграцією з фабрикою JLCPCB. Нижче було наведено недоліки та переваги цієї середи розробки електроніки.

Хмарна архітектура та колаборація:

- працює в браузері без встановлення клієнта, що дозволяє команді одночасно редагувати проєкт у реальному часі незалежно від ОС або місця розташування;
- автоматичне збереження на сервері та версіонування проєктів підвищують надійність і зручність командної роботи.

Бібліотека компонентів:

- поєднання трьох основних репозиторіїв: офіційні LCSC-компоненти, системні (open-source) бібліотеки та user-contributed каталоги;
- наявність неофіційних модулів, серед яких DevEBox STM32H7XX, дозволяє миттєво підключати специфічні плати та мікроконтролери без ручного створення footprint/symbol.

Інтеграція з виробництвом:

- пряме замовлення PCB-прототипів та плат SMT-збирання через JLCPCB із редактора EasyEDA одним кліком;
- генерація Gerber, BOM, pick-and-place файлів і готових вихідних комплектів для виробництва.

Професійні інструменти трасування:

- підтримка багатошарових плат, автотрасування з можливістю ручного редагування, автоматична перевірка DRC за правилами ДСТУ;
- 3D-перегляд готового проекту для оцінки компонентного розміщення та механічних вимог.

Недоліки:

- обмежена кількість приватних проектів у безкоштовній редакції; для розширення необхідна платна підписка;
- повільна робота великого проекту через залежність від швидкості інтернет-з'єднання та браузера;
- часто критикують довгий доступ до user-contributed бібліотек у Pro-версії та відсутність можливості повністю приватизувати репозиторій.

### 3.1.2 Altium Designer

Altium Designer – провідна у світі комерційна ECAD/PCB платформа з єдиним середовищем розробки, що охоплює схеми, трасування, симуляцію та MCAD-інтеграцію

Переваги:

- високопродуктивні алгоритми авто-трасування (ActiveRoute, push-and-shove) і глобальної оптимізації доріжок;
- підтримка rigid-flex плат, ECAD-MCAD co-design, реального часу оцінка вартості, управління постачанням (Altium 365);
- глибока інтеграція бібліотек, PLM-інструменти та командна робота через Altium 365 Workspace.

Недоліки:

- висока вартість ліцензії (від кількох тисяч USD на рік) та складний інтерфейс для новачків;
- застосунок лише для Windows, потребує продуктивного ПК.

Висновок по Altium:

Altium Designer – вибір провідних інженерних команд для великих проектів, однак його висока вартість і складність виходять за межі завдань

студентського прототипування та швидкого макетування.

### 3.1.3 KiCad

KiCad – безкоштовна, відкрита EDA-платформа з розвитком підтримуваним CERN та спільнотою, що включає схему, PCB, SPICE-симуляцію і 3D-перегляд.

Переваги:

- повна відсутність ліцензійних витрат (GPL-3.0), активні оновлення і велика кількість сторонніх бібліотек (SnapEDA, Digi-Key);
- підтримка багат шарових плат, детальна DRC і можливість імпорту з Altium/Eagle/EasyEDA.

Недоліки:

- відсутність інтеграції “одним кліком” із виробництвом – необхідно експортувати Gerber та окремо завантажувати на фабрику;
- більш довгий стартовий час налаштування бібліотек, складний інтерфейс для недосвідчених користувачів.

Висновок по KiCad:

KiCad ідеально підходить для відкритих проєктів та DIY-спільнот, проте не забезпечує той рівень швидкості замовлення та user-contributed компонентів, який потрібен для структурування макету DevEBox.

## 3.2 Використані компоненти та їх призначення

У цьому підрозділі наведено перелік основних електронних компонентів функціонального макету плати керування світлофорами на базі DevEBox STM32H7XX\_M, коротко описано їхні характеристики, логіку застосування та типовий розрахунок номіналів (таблиця 3.1).

Таблиця 3.1 – Перелік основних електронних компонентів функціонального макету

№	Електронний компонент	Призначення
1	C1-C8 (конденсатори, 100 nF, SMD 0805)	локальні фільтри живлення (bypass) на входах VDD MCU для згладжування високочастотних шумів
2	C11-C13, C20 (конденсатори, 10 $\mu$ F, SMD 1206)	буферні конденсатори на лінії +3 V3 для компенсації стрибків навантаження (Ethernet, SDMMC)
3	C14 (конденсатор, 2,2 $\mu$ F, SMD 0805)	фільтрація опорної напруги REF-DAC
4	C15, C16 (конденсатори, 33 pF, SMD 0805)	корекція ліній сигналів високошвидкісних інтерфейсів (SDMMC, SPI)
5	C18, C29, C30 (конденсатори, 10 nF, SMD 0805)	додаткові фільтри живлення периферійних інтерфейсів. C19 (конденсатор, 4,7 $\mu$ F, SMD 0805), фільтрація лінії +12 V для живлення LED1 та індуктивних петель
6	D1-D7 (діоди, 1N4148W, SOD-123)	захист і логічне об'єднання ліній SiSi (Signalsicherung) для аварійного миготіння жовтим
7	D13 (стабілітрон, BZX84C3V3, SOT-23)	стабілізатор 3,3 V для живлення I <sup>2</sup> C та SPI інтерфейсів
8	JP1-JP6 (джампери, перемички)	налаштування каналів Watchdog (які LED блиматимуть при відмові MCU)
9	K1 (реле, НК115FD-DC5V-SG)	аварійне відключення лінії живлення світлофорів при критичній помилці MCU
10	L1 (індуктор, НН-1M1608-12ІТ, 12 $\mu$ H)	шунтування імпульсних перешкод від індукційних петель
11	LED1 (світлодіод, 12 V)	індикація відгуку індуктивних петель (активний 12 V)
12	LED2	аварійне миготіння жовтим (линяння)
13	LED3	сигнал переходу управління до Watchdog-MCU (AT89C4051)
14	LED4	тактуючий імпульс 1 Hz від головного CPU

Продовження таблиці 3.1

№	Електронний компонент	Призначення
15	Q1–Q6 (транзистори, BCX70H,215, SOT-23)	комутація світлодіодного навантаження через PCA9555D
16	R3–R18, R37, R50, R56, R57 (резистори, UniOhm 1206- series)	обмеження струму світлодіодів та pull- up/pull-down для логічних ліній
17	RN1, RN2 (мережеві резистори, 4×4,7 kΩ)	групові pull-up для I <sup>2</sup> C та CD74HC245
18	U1 (IC, DevEBox-STM32H743VI)	основний модуль MCU STM32H743VIT6 з вбудованим USB- DFU, microSD-слотом, Ethernet, CAN- FD
19	U2–U9 (IC, PCA9555D,112)	16-канальні I <sup>2</sup> C-експандери портів для індикації світлофорів
20	U6 (IC, CD74HC245M96)	двонаправлений буфер даних для LCD/LED індикації
21	U15 (IC, AT89C4051-24SU)	автономний Watchdog-MCU для аварійного керування
22	U17 (IC, W5500, LQFP-48)	Ethernet-контролер з апаратним TCP/IP стеком
23	U18 (роз'єм, RJMG163118101NR)	інтерфейс RS-232 для діагностики
24	X1 (кварц, ABMM-6.000MHZ- B2-T)	тактовий резонатор 6 MHz для Watchdog-MCU
25	X3 (кварц, X503225MSD4SC, 32.768 kHz)	низькочастотний резонатор для RTC

Це забезпечує чітку трасування компонентів за їхніми позначеннями та дозволяє безпомилково інтерпретувати кожен елемент схеми під час документування й виготовлення за стандартами ДСТУ.

Приклад розрахунку для C1-C8:

при  $ESR \approx 0,1 \Omega$  та  $C = 100 \text{ nF}$  частота зрізу  $F_c = 1/(2\pi \cdot ESR \cdot C) \approx 15,9 \text{ MHz}$ ,  
що ефективно гасить шуми шин живлення;

Приклад розрахунку для C14:

$$F_c = \frac{1}{2\pi R_{in} C} = \frac{1}{2\pi \times 1 \text{ k}\Omega \times 2.2 \text{ }\mu\text{F}} = 72 \text{ Hz}$$

Приклад розрахунку для LED1:

$$R = \frac{12\text{V} - 2\text{V}}{10\text{mA}} = 1\text{k}\Omega$$

### 3.3 Тестування критичних протоколів

Для забезпечення належної функціональності та надійності розробленої системи керування міським трафіком необхідно виконати тестування основних комунікаційних інтерфейсів мікроконтролера STM32H743VIT6. У підкресленні критичності цих протоколів полягає те, що від їхньої злагодженої роботи залежить коректний обмін даними між центральним процесором і периферійними модулями, а отже й успішне виконання алгоритмів регулювання світлофорів. У цьому розділі наведено обґрунтування важливості протоколів UART, SPI, I<sup>2</sup>C та SDIO, а також описано принципи їхнього тестування в умовах реального стенда з використанням макетної плати (breadboard) та реальних периферійних пристроїв.

Критичні комунікаційні протоколи забезпечують взаємодію мікроконтролера з такими компонентами, як:

- UART (Universal Asynchronous Receiver/Transmitter) – для передачі діагностичних та сервісних повідомлень, а також підключення зовнішніх консолей і модулів трансиверів (наприклад, MAX3232IPW);

- SPI (Serial Peripheral Interface) – для високошвидкісної передачі даних між MCU та Ethernet-контролером W5500, а також резервного інтерфейсу для SD-карти;

– I<sup>2</sup>C (Inter-Integrated Circuit) – для зв'язку з розширювачами портів PCA9555D, зовнішніми дисплеями, сенсорами й іншими малошвидкісними пристроями;

– SDIO (Secure Digital Input/Output) через контролер SDMMC1 – для роботи з microSD-картою, що використовується для збереження логів і архівів відео з камер.

Коректна робота кожного з перелічених інтерфейсів є необхідною умовою забезпечення реального часу виконання алгоритмів фазового керування та аварійних сценаріїв. Часто взаємодія з периферією у режимі максимальних швидкостей створює суворі вимоги до таймінгів сигналів, напруги, електричних характеристик ліній і опорів. Тому перевірка лише на програмній емуляції або в онлайн-симуляції не може гарантувати відсутність помилок у фізичній реалізації. З цієї причини тестування проводиться на реальних макетних платах (breadboard) із справжніми компонентами, джерелом живлення та вимірювальними приладами (осцилограф, логічний аналізатор).

Для тестування було використано логічний аналізатор Analog Discovery 2 ( рис. 3.2) від фірми Digilent та осцилограф з аналізатором протоколів.

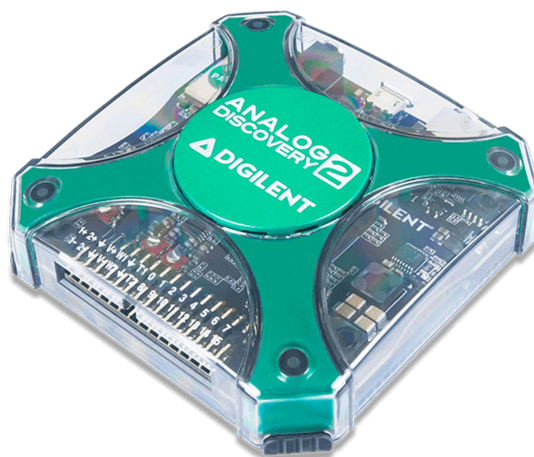


Рис 3.2 - Зовнішній вигляд логічного аналізатора Analog Discovery 2 від Digilent.

Для забезпечення коректної роботи інтерфейсів UART, SPI, I<sup>2</sup>C та SDIO на апаратному макеті була реалізована та перевірена прошивка в середовищі STM32CubeIDE.

У створеному в STM32CubeIDE проєкті модулі поділено на групи:

- Core/Inc – заголовочні файли інтерфейсів і middleware:
- stm32h7xx\_hal\_conf.h – конфігурація HAL (активація модулів UART, SPI, SDMMC, I<sup>2</sup>C);
- stm32h7xx\_it.h – обробники переривань;
- w5500.h, wizchip\_conf.h, socket.h, dhcp.h – бібліотека WIZnet для роботи з Ethernet (SPI);
- i2c.h, ssd1306.h, fonts.h – драйвер SSD1306 OLED-дисплея по I<sup>2</sup>C з підтримкою виводу тексту та шрифтів;
- spi.h – обгортувач HAL для SPI1/2, використовується як для W5500, так і резервно для SD-карти;
- usart.h – драйвер UART для COM-порту та моніторингу;
- sdmmc.h – обгортка HAL SDMMC1 для роботи з microSD через SDIO;
- gpio.h, memorymap.h – низькорівневі конфігурації виводів та організації пам'яті.
- Core/Src – вихідні файли реалізації:
- main.c – початкове налаштування периферії, виклик тестових функцій;
- stm32h7xx\_hal\_msp.c – ініціалізація GPIO, тактування, переривань;
- w5500.c, socket.c, dhcp.c – стек W5500: налаштування SPI, встановлення IP через DHCP, TCP/UDP-сокети;
- i2c.c, ssd1306.c, fonts.c – I<sup>2</sup>C-інтерфейс: ініціалізація шини, запис даних на OLED;
- spi.c – загальні SPI-функції (HAL\_SPI\_Transmit/Receive);
- usart.c – налаштування UART (115200 8N1), функції передачі та прийому;
- sdmmc.c – робота з microSD: ініціалізація SDMMC1, FATFS-монтування, читання/запис файлів;

– syscalls.c, system.c, system\_stm32h7xx.c – забезпечення роботи функцій стандартної бібліотеки й системного такту.

### 3.3.2 Налаштування HAL і middleware

- В CubeMX (через .ioc файл) активовано:
- USART2: для тесту передавання повідомлень на ПК;
- SPI1 (Master mode ) з прескалером в 32 (швидкість 2.25 мб/с), що забезпечує  $SCK \leq 60$  MHz для W5500;
- I<sup>2</sup>C1 : для SSD1306 OLED;
- SDMMC1 (4-bit mode) з DMA для швидкого доступу до microSD;
- FreeRTOS (опціонально) – для організації тестових задач;
- згенеровано код ініціалізації периферії у MX\_\*\_Init() функціях, які викликаються в main.c.

### 3.3.3 Методика тестування:

#### 3.3.3.1 USART

Мета: перевірити коректність ініціалізації та швидкість без втрат.

Процедура: у main.c після старту HAL викликається usart\_test() – передача системного тикуну процесора.

На ПК через USB-UART адаптер (CP2102) прийом даних аналізується в терміналі (PuTTY) з підрахунком помилок.

#### 3.3.3.2 SPI + W5500

Мета: перевірити взаємодію MCU з Ethernet-контролером.

Процедура: ініціалізація SPI1 та W5500 через wizchip\_init() у w5500.c.

Отримання IP через DHCP (dhcp\_run()), перевірка IP-адреси на OLED.

Відправка ICMP Echo-request (socket\_ping()), аналіз відповіді.

Осцилографом заміряються SCK-імпульси, MOSI/MISO-фази.

### 3.3.3.3 I<sup>2</sup>C + OLED

- Мета: перевірити роботу шини I<sup>2</sup>C та відображення інформації.
- Процедура:
  - Ініціалізація I<sup>2</sup>C4 та SSD1306 (ssd1306\_Init()).
- Вивід повідомлення з IP адресою та монтуванням SD картки.

### 3.3.3.4 SDIO (SDMMC1)

- Мета: перевірити швидкість та стабільність роботи з microSD.
- Процедура:
  - sdmmc\_test() в sdmmc.c монтує FATFS та створює файл test.txt розміром 10 МБ.
  - в перемінну buff були виведені значення які були раніше записані в файл test.txt.
  - через OLED дисплей було виведено повідомлення про успішне монтування картки (“SD MOUNT OK“).

## 3.4 Трасування друкованої плати

У процесі розробки РСВ для функціонального макету системи керування світлофорами на базі DevEBox STM32H7XX\_M дотримано основні правила трасування відповідно до рекомендацій виробника STMicroelectronics.

Нижче наведено ключові рішення та обґрунтування розміщення компонентів і вибору ширини доріжок:

- розташувати конденсатори розв’язки;
- декілька 100 nF (C1–C8) та буферні 10 μF (C11–C13, C20) конденсатори розв’язки розмістити максимально близько до живильних висновків MCU (VDD, VDD\_IO) і регуляторів напруги;
- мінімізація індуктивних та резистивних падінь напруги на високих частотах, забезпечуючи стабільність живлення ядра та периферії при

швидких перемиканнях;

- групувати компоненти за функціями;
- всі логічні буфери (PCA9555D, CD74HC245), мікроконтролер Watchdog (AT89C4051) та живильні стабілізатори розмістити в окремій області праворуч для зменшення перешкод від потужних ліній 12 V;
- блок Ethernet (W5500) та розширювачі портів PCA9555D розташувати поруч із STM32 для коротких SPI- та I<sup>2</sup>C-ліній;
- реле K1 та індикаторні LED закріпити на периферії плати з урахуванням простору для підключення зовнішніх кабелів;
- розподілити лінії живлення;
- лінії +12 V та +5 V виконати шириною 24 mil ( $\approx 0,6$  мм) – для пропуску струмів до 1 A без значного нагрівання;
- лінія +3.3 V для живлення MCU та логіки виконати шириною 16 mil ( $\approx 0,4$  мм), що достатньо при струмах до 500 mA і відповідає рекомендаціям RM0433;
- розділення «землі»;
- використати двошарову плату з окремими зонами GND для силової (12 V/5 V) та цифрової (3.3 V) частин;
- об'єднання відбувається в єдиній точці «зірка» біля входу живлення для запобігання утворенню петльових струмів та зменшення шуму;
- трасувати високошвидкісні сигнали;
- лінії SPI1 (SCK, MOSI, MISO) та SDMMC1 трасувати максимально коротко та прямолінійно, з оглядом на однакову довжину для мінімізації затримок і відбиттів;
- резистори серії RN1, RN2 для I<sup>2</sup>C-пул-ап розмістити біля роз'єму дисплея та PCA9555D, щоб забезпечити коректні часи встановлення сигналів;
- розмістити транзисторний блок;
- Q1–Q6 (BCX70H) для комутації світлофорів ставити поруч із відповідними розширювачами портів PCA9555D (U2–U9) для мінімального шляху від виходів порту до бази транзисторів;

– оцінити 3D-розміщення.

На рис. 3.3 виконано перевірку 3D-моделі, що підтвердила відсутність механічних конфліктів між компонентами та достатній зазор для монтажу під DevEBox-модуль і роз'єм RS-232.

Дотримання вищеописаних принципів трасування гарантує стабільну роботу плати в умовах промислових перешкод, забезпечує коректні електричні параметри інтерфейсів та спрощує виготовлення й подальший монтаж.

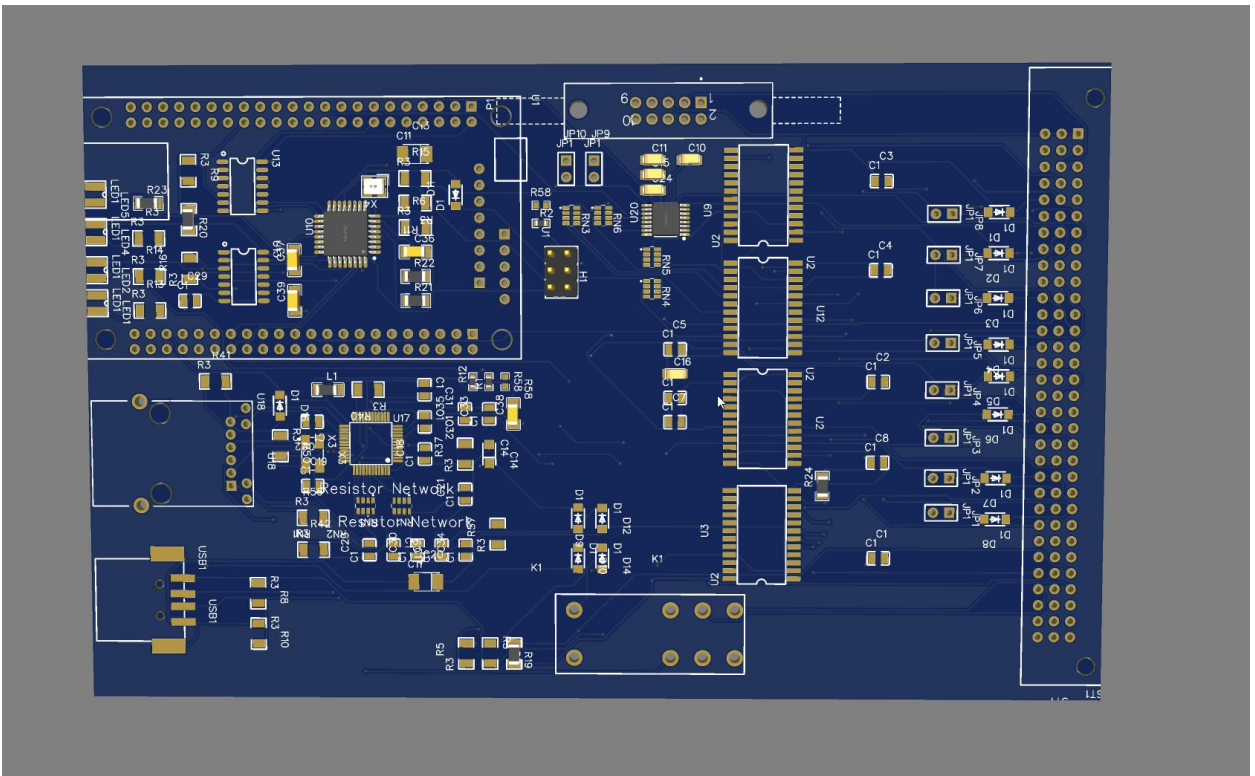


Рисунок 3.3 – Зовнішній вигляд відтрасованої плати в 3D

### 3.4 Охорона праці під час розробки, налагодження й експлуатації автоматизованої системи

Охорона праці є невід’ємною складовою безпеки під час розробки, налагодження й експлуатації автоматизованої системи керування міським трафіком на базі STM32H743VIT6 [36].

Урахування вимог безпеки дозволяє мінімізувати ризики травм,

ураження електричним струмом та негативного впливу на здоров'я обслуговуючого персоналу:

- організація робочого місця;
- усі операції з монтажу та паяння друкованих плат слід виконувати в добре провітрюваному приміщенні або за допомогою витяжної установки для видалення флюсу та парів припою;
- робочі поверхні мають бути заземлені через антистатичні покриття, а персонал – забезпечений екранізованими браслетами та килимками для запобігання ESD-пошкодженню чутливих компонентів;
- електробезпека;
- у приміщенні необхідно встановити окремі вимикачі для аварійного знеструмлення системи: головний (відключення всіх ліній живлення +12 V, +5 V, +3.3 V) та локальні (відключення лише силових реле й індуктивних елементів);
- усі кабельні колодки та роз'єми (RS-232, Ethernet, I<sup>2</sup>C-лінії) мають бути заземлені через загальну систему «зірка», яка запобігає витоку струму та електромагнітним завадам;
- технічне обслуговування;
- регулярний візуальний огляд і очищення плати від пилу та вологи – не рідше ніж раз на 6 місяців;
- контроль стану теплоотводів і радіаторів Ethernet-контролера W5500, а також перевірка контактів на роз'ємах живлення і периферії для уникнення перегріву та погіршення контактів;
- системи захисту;
- автоматичне відключення – у разі виходу з ладу основного MCU або Watchdog-мікроконтролера спрацьовує захист SiSi, і всі світлофори переходять у жовтий миготливий режим;
- захист від короткого замикання – запобіжники на лініях +12 V і +5 V, а також РТС-резистори на критичних силових доріжках для обмеження струму при замиканні;

– система пожежогасіння – у приміщенні встановлюють вогнегасники класу Е (для електрообладнання) і датчики диму з автоматичним сповіщенням;

– вентиляція та фільтрація – забезпечується припливно-витяжна вентиляція з НЕРА-фільтрами для контролю вологості та видалення шкідливих парів, які можуть виникнути при паянні;

– сигналізація про відмови – світлова (червоний індикатор на щиті керування) та звукова (сирена 85 дБ) система оповіщення обслуговуючого персоналу про критичні збої чи перевантаження в роботі плати.

Дотримання цих заходів гарантує не лише безпеку працівників під час розробки й технічного обслуговування, але й надійну експлуатацію автоматизованої системи керування міським трафіком у критичних та аварійних режимах [36].

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було здійснено дослідження, метою якого було розроблення автоматизованої системи контролю міського трафіку на базі мікроконтролера STM32H743VIT6.

Основний акцент було зроблено на проєктуванні апаратно-програмного комплексу: від вибору MCU та середовища розробки до трасування друкованої плати, а також тестування критичних комунікаційних протоколів у реальних умовах.

На основі аналізу сучасних стандартів DIN VDE V 0823-500, RiLSA та європейських норм ITS було обґрунтовано вибір платформи DevEBox STM32H7XX\_M і середовища EasyEDA для швидкого макетування й інтеграції з виробництвом.

Розроблено структурну схему, розміщено та трасовано компоненти з урахуванням вимог електробезпеки, шумозаглушення та теплорозподілу. Була реалізована прошивка в STM32CubeIDE із підтримкою UART, SPI/W5500, I<sup>2</sup>C/SSD1306 та SDIO/FATFS, яку успішно протестовано на реальному макеті із використанням осцилографа, логічного аналізатора й стендового обладнання.

В результаті виконаної роботи створено працездатний прототип системи автоматизації світлофорів із можливістю аварійного переходу в жовте миготіння, мережевою комунікацією Ethernet, індикацією стану та архівацією даних на microSD.

Розроблене рішення демонструє високу гнучкість, відповідність міжнародним і національним стандартам та потенціал для впровадження у міській інфраструктурі з метою підвищення пропускної здатності перехресть і безпеки руху.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Волоніхін В. Д. Аналіз можливості інтеграції мікроконтролера на базі STM32 в сучасні прилади керування світлофорами/ В. Д. Волоніхін // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED-2025) : збірник студентських наукових статей. Харків : ХНУРЕ, 2025. Вип. 1. С. 222–228.
2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
3. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, В.А. Андрусевич, О. В. Токарева, С. П. Новоселов, О. В. Сичова. Харків : Видавництво Іванченка І. С., 2022. 151 с.
4. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарева, С.П. Новоселов, О.В. Сичова. Харків: ХНУРЕ, 2023. 64 с.
5. STMicroelectronics. STM32H743VI – High-performance MCU with ARM® Cortex®-M7. [Електронний ресурс]. Режим доступу: <https://www.st.com/en/microcontrollers-microprocessors/stm32h743vi.html>
6. STMicroelectronics. Datasheet STM32H743xI/G (PDF). [Електронний ресурс]. Режим доступу: <https://www.st.com/resource/en/datasheet/stm32h743vi.pdf>

7. STMicroelectronics. STM32H7 Series – Products. [Электронный ресурс]. Режим доступа: <https://www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html>
8. SWARCO FUTURIT. Inductive Loop Detectors. [Электронный ресурс]. Режим доступа: <https://www.swarco.com/products/urban-traffic-solutions/loop-detectors>
9. FLIR Systems. TrafiOne Thermography Cameras. [Электронный ресурс]. Режим доступа: <https://www.flir.com/products/trafione/>
10. RTB Safe Traffic. APS Pedestrian Push Buttons. [Электронный ресурс]. Режим доступа: <https://www.rtbtraffic.com/en/products/push-buttons/>
11. STMicroelectronics. STM32CubeMX. [Электронный ресурс]. Режим доступа: <https://www.st.com/en/development-tools/stm32cubemx.html>
12. STMicroelectronics. STM32CubeIDE. [Электронный ресурс]. Режим доступа: <https://www.st.com/en/development-tools/stm32cubeide.html>
13. Wikipedia. Vehicle-to-Everything (V2X). [Электронный ресурс]. Режим доступа: <https://en.wikipedia.org/wiki/Vehicle-to-everything>
14. NXP Semiconductors. S32K344 Automotive MCU. [Электронный ресурс]. Режим доступа: <https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/s32-k-automotive-cortex-m-based-microcontrollers:S32K344>
15. Texas Instruments. Hercules TMS570LC4357. [Электронный ресурс]. Режим доступа: <https://www.ti.com/product/TMS570LC4357>
16. Infineon Technologies. AURIX™ TC275. [Электронный ресурс]. Режим доступа: <https://www.infineon.com/cms/en/product/microcontroller/32-bit-tricore-microcontroller-aurix/aurix-tricore-tc2xx-series/tc275t/>
17. Renesas Electronics. R-Car® H3e. [Электронный ресурс]. Режим доступа: <https://www.renesas.com/us/en/products/mpc/automotive-r-car/r-car-h3>
18. Microchip Technology. SAMA5D27 Single-Board Computer. [Электронный ресурс]. Режим доступа: <https://www.microchip.com/wwwproducts/en/SAMA5D27>

19. Xilinx (AMD). Zynq®-7000 XC7Z010. [Електронний ресурс]. Режим доступу: <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>
20. ISO. ISO/TS 14812:2022 Intelligent transport systems – Vocabulary. [Електронний ресурс]. Режим доступу: <https://www.iso.org/standard/62048.html>
21. CEN/TC 278. Intelligent Transport Systems Standards. [Електронний ресурс]. Режим доступу: [https://standards.cen.eu/dyn/www/f?p=204:110:0:::FSP\\_PROJECT,FSP\\_ORG\\_ID:46440,2612](https://standards.cen.eu/dyn/www/f?p=204:110:0:::FSP_PROJECT,FSP_ORG_ID:46440,2612)
21. Institute of Transportation Engineers. Traffic Control Devices Handbook, 8th ed., ITE 2020. [Електронний ресурс]. Режим доступу: <https://www.ite.org/publications/traffic-control-devices-handbook/>
22. Wikipedia. SPICE (Simulation Program with Integrated Circuit Emphasis). [Електронний ресурс]. Режим доступу: [https://en.wikipedia.org/wiki/SPICE\\_\(simulation\\_program\)](https://en.wikipedia.org/wiki/SPICE_(simulation_program))
23. Wikipedia. Printed circuit board. [Електронний ресурс]. Режим доступу: [https://en.wikipedia.org/wiki/Printed\\_circuit\\_board](https://en.wikipedia.org/wiki/Printed_circuit_board)
24. Wikipedia. Electronic design automation. [Електронний ресурс]. Режим доступу: [https://en.wikipedia.org/wiki/Electronic\\_design\\_automation](https://en.wikipedia.org/wiki/Electronic_design_automation)
25. Wikipedia. Design rule checking. [Електронний ресурс]. Режим доступу: [https://en.wikipedia.org/wiki/Design\\_rule\\_checking](https://en.wikipedia.org/wiki/Design_rule_checking)
26. Europan.ua. Міжнародна стандартизація. Німецькі стандарти DIN та VDE. [Електронний ресурс]. Режим доступу: <https://europan.ua/news/mezhdunarodnaja-standartizacija-nemeckie-standarty-din-i-vde/>
27. DIN e.V. DIN V VDE V 0823-500: Richtlinien für Lichtsignalanlagen. [Електронний ресурс]. Режим доступу: <https://www.din.de>
28. STMicroelectronics. Reference Manual RM0433 – STM32H7 Series, advanced ARM®-based 32-bit MCUs. [Електронний ресурс]. Режим доступу: [https://www.st.com/resource/en/reference\\_manual/rm0433-stm32h7-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/rm0433-stm32h7-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf)
29. STMicroelectronics. User Manual UM2564 – STM32Cube HAL and Low-Layer Drivers. [Електронний ресурс]. Режим доступу:

[https://www.st.com/resource/en/user\\_manual/um2564-stm32cube-hal-and-lowlayer-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um2564-stm32cube-hal-and-lowlayer-drivers-stmicroelectronics.pdf)

30. WIZnet. W5500 Hardwired TCP/IP Embedded Ethernet Controller. [Електронний ресурс]. Режим доступу: <https://www.wiznet.io/product-item/w5500/>

31. ChaN. FatFs – Generic FAT file system module for small embedded systems. [Електронний ресурс]. Режим доступу: [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)

32. NXP Semiconductors. PCA9555 – I<sup>2</sup>C-bus I/O Port Expander. [Електронний ресурс]. Режим доступу: <https://www.nxp.com/docs/en/datasheet/PCA9555.pdf>

33. Texas Instruments. CD74HC245 – Octal Bus Transceiver. [Електронний ресурс]. Режим доступу: <https://www.ti.com/lit/ds/symlink/cd74hc245.pdf>

34. Microchip Technology. AT89C4051 – 4-Kbyte Flash, 128-byte RAM, 20-pin Microcontroller. [Електронний ресурс]. Режим доступу: <https://ww1.microchip.com/downloads/aemDocuments/documents/ProductDocuments/DataSheets/AT89C4051-24SU.pdf>

35. Abracon LLC. ABMM-6.000MHZ-B2-T – SMD Crystal Resonator. [Електронний ресурс]. Режим доступу: [https://abracon.com/Crystal\\_Current\\_Controlled\\_Crystals/ABMM.pdf](https://abracon.com/Crystal_Current_Controlled_Crystals/ABMM.pdf)

36. Босов Є. П., Жесан Р. В., Каліч В. М., Голик О. П., Зубенко В. О. Охорона праці при проектуванні систем автоматизації виробництва : навч. посіб. 2-е вид., перероб. і доп. Кропивницький : ЦНТУ, 2022. 208 с.