

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ОСАДЧУК Михайло Андрійович

**«Алгоритми оптимізації транспортних потоків на
автострадах / Algorithms for optimizing traffic flows
on highways»**

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІм-22
М.А. Осадчук

Науковий керівник:
к.т.н. Н.Я. Савка

Кваліфікаційну роботу допущено
до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
_____ О.Л. Дубчак

Тернопіль – 2022

ЗМІСТ

Вступ.....	7
1 Аналіз засобів реалізації системи управління транспортними потоками	10
1.1 Аналіз засобів регулювання дорожнього руху	10
1.2 Аналіз методів автоматизації керування дорожнім рухом	20
1.3 Аналіз засобів реалізації системи управління світлофорами	20
1.4 Постановка задачі кваліфікаційної роботи	29
1.5 Висновки до розділу	30
2 Алгоритм автоматизованого регулювання дорожнього руху	31
2.1 Системи керування дорожнім рухом	31
2.2 Методи оптимізації регулювання дорожнім рухом	37
2.3 Еврастичні алгоритми керування транспортними потоками.....	40
2.4 Система оптимізації транспортних потоків.....	46
2.5 Алгоритм оптимізації транспортних потоків на автострадах.....	49
2.6 Висновки до розділу	53
3 Реалізація алгоритму оптимізації транспортних потоків на автострадах.	54
3.1 Структура програмного модуля системи виявлення та підрахунку транспортного на автострадах	54
3.2 Реалізація алгоритму виявлення та підрахунку транспорту на автострадах.....	57
3.3 Модель підсистеми автоматизованого управління світлофорами на основі мереж Петрі	63
3.4 Модель системи автоматизованого управління світлофорами.....	64
3.5 Програмно-апаратна реалізація системи автоматизованого управління світлофорами	68
3.6 Висновки до розділу	75
Висновки.....	76
Список використаних джерел.....	77
Додаток А Світлокопії публікацій	82

ВСТУП

Збільшення заторів призводить до численних втрат ресурсів країн через брак бензину, виснаження автомобілів і запізнень робітників, на додаток, втома, як громадян, так і співробітників дорожньої поліції. Багато проблем із звичайним контролером світлофора, великий затор – найважливіша проблема, тому що ніколи неможливо оцінити рівень пробки в кожному напрямку і неможливо встановити час затримки.

Інша проблема виникає, коли затору немає, але очікування все ще триває. Вирішення цих проблем полягає в тому, щоб визначити рівень застрягання і встановити час затримки. На додаток до проблем із затримкою часу існує проблема вчасного прибуття аварійних машин, таких як швидка допомога, поліція та пожежна команда. Ця проблема вимагає оцінки співробітника поліції, оскільки виникає необхідність в ручному управлінні світлофором. Пробка може зменшитися, якщо водій отримає інформацію про рівень скупчення транспортних засобів на дорозі до того, як досягне її, і вибере інший шлях.

Розробка моделі системи автоматизованого регулювання дорожнього руху, яка дозволить враховувати поточну ситуацію в дорожній мережі у режимі реального часу є важливою та актуальною задачею для покращення регулювання дорожнього руху та зменшення часу, який витрачає окрема людина кожен день в дорозі, використовуючи наземний транспорт.

Зважаючи на це, у роботі буде запропоновано модель мікроконтролерної системи автоматизованого управління світлофорами на основі аналізу руху автотранспорту на основі методів комп'ютерного зору, з використанням ультразвукового датчика відстані та реалізацію його показників на базі мікроконтролера Arduino.

Мета та завдання дослідження. Зважаючи на вищезазначене, метою кваліфікаційної роботи є розробка алгоритму оптимізації транспортних потоків на автострадах.

Для досягнення мети роботи слід виконати такі задачі:

- провести аналіз засобів регулювання дорожнього руху;
- охарактеризувати методи автоматизації керування дорожнім рухом;
- розробити алгоритм оптимізації транспортних потоків на автострадах;
- удосконалити схему оптимізації транспортних потоків на автомобільних магістралях;
- розробити програмно-апаратний комплекс для реалізації алгоритму оптимізації транспортних потоків на автострадах;
- провести експериментальні дослідження ефективності розробленого алгоритму оптимізації транспортних потоків на автострадах;
- удосконалити автоматизовану систему оптимізації транспортних потоків;

Об'єкт дослідження. Процеси керування дорожнім рухом.

Предмет дослідження. Алгоритм оптимізації транспортних потоків на автострадах.

Методи дослідження. Методи автоматизації для побудови алгоритму керування дорожнім рухом, методи класифікації для розробки системи управління транспортними потоками, комп'ютерного моделювання для реалізації системи управління світлофорами на основі ультразвукового датчика руху.

Наукова новизна одержаних результатів.

1. Розроблено алгоритм автоматизованого регулювання дорожнього руху, який, на відміну від існуючих, базується на датчачі руху.
2. Удосконалено автоматизовану систему оптимізації транспортних потоків на автострадах.

Практичне значення отриманих результатів. На основі розроблених алгоритмів із використанням мови програмування C у інтегрованому середовищі розробки Arduino IDE розроблено програмно-апаратне забезпечення для оптимізації автоматизованого регулювання дорожнього руху на автострадах.

Публікації та апробація результатів. Основні результати дослідження опубліковано на XIV Всеукраїнській науково-практичній інтернет-конференції «Foreign languages in use: academic and professional aspects» [] та IV Міжнародній

науково-практичній конференції «Discussion and development of modern scientific research»[]. Копії публікацій наведено у додатку А.

У першому розділі проведено аналіз засобів реалізації системи управління транспортними потоками описано основні методи їх класифікації, проаналізовано засоби реалізації системи управління світлофорами. Досліджено існуючі автоматизовані системи управління транспортними потоками, виокремлено основні їх недоліки. Здійснено постановку задачі на кваліфікаційну роботу на основі проведеного аналізу.

У другому розділі описано алгоритми автоматизованого регулювання дорожнього руху, етапи роботи системи й зазначено, що основними є – попереднє опрацювання сигналу давача руху. Проаналізовано методи оптимізації транспортних потоків. Розроблено алгоритм виявлення та підрахунку автотранспорту, а також алгоритм автоматизованого управління світлофором.

У третьому розділі описано програмне середовище для розробки програмної системи виявлення та підрахунку автотранспорту. Розроблено й описано діаграму варіантів використання програмної системи. Розроблено апаратний засіб оптимізації транспортних потоків на автострадах на основі алгоритму автоматизованого управління світлофором. Проведено аналіз точності роботи програмного засобу виявлення та підрахунку автотранспорту.

У додатках представлено копії публікацій результатів кваліфікаційної роботи.

1 АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ ТРАНСПОРТНИМИ ПОТОКАМИ

1.1 Аналіз засобів регулювання дорожнього руху

Для створення автоматизованої системи світлофорами проаналізуємо засоби регулювання дорожнього руху та дамо означення основним поняттям для створення алгоритму оптимізації транспортних потоків на автострадах.

Рух – це переміщення людей і товарів з одного місця в інше. Переміщення зазвичай відбувається вздовж певного об'єкта або шляху, який можна назвати напрямком, або дорогою [27, 42].

Трафік розвивається через необхідність переміщення людей і товарів з одного місця в інше. Таким чином, рух ініціюється через рішення, прийняті людьми перевозити себе чи інших з одного місця в інше для участі в діяльності в другому місці або переміщення товарів у місце, де вони мають вищу цінність. Таким чином, транспортні потоки принципово відрізняються від інших галузей інженерії та фізичних наук (таких як рух електронів у дроті), оскільки вони головним чином керуються та визначаються законами людської поведінки [42].

На найнижчому рівні контроль дорожнього руху включає планування вулиць для задоволення різноманітних потреб подорожей у регіоні. На автомагістралях або на швидкісних дорогах рух транспорту здійснюється на високій швидкості, магістральними вулицями пересувається рух всередині та вздовж міських районів, місцеві вулиці забезпечують низьку швидкість, але доступ до багатьох місцевих напрямків. Ієрархія вулиць, які працюють на різних рівнях швидкості та забезпечують різні рівні доступу, утворюють основу, на якій розвиваються проблеми регулювання дорожнього руху. Тривалі затримки та часті аварії є поширеними наслідками неадекватного планування доріг, що призводить до недостатньої кількості доріг для задоволення потреб подорожей. Тому контроль дорожнього руху на таких ділянках особливо важливий, оскільки це може суттєво виправити вище перераховані недоліки [42].

Контроль дорожнього руху на самому елементарному рівні досягається за допомогою системи знаків, сигналів і розмітки. Розроблені інженерні стандарти використовуються, щоб гарантувати, що пристрої регулювання дорожнього руху передають ясне та просте значення автомобілісту. Потрібна навчальна програма, яка впроваджується через органи видачі водійських прав, щоб гарантувати, що ті, хто керує транспортними засобами, розуміють правила дорожнього руху та дії, які вони зобов'язані виконувати, коли присутній певний контрольний пристрій [8].

Для регулювання трафіку використовуються технічні засоби організації дорожнього руху транспортних та пішохідних потоків. Регулювання (від латинської «regula» – правило) відповідає за організацію та безпеку транспортних потоків і пішоходів та за підтримку певного рівня їх показників [8].

Слід зазначити, що технічні засоби регулювання розділяються на декілька типів.

Пристрої та знаки для інформування учасників дорожнього руху:

- дорожні знаки, а також покажчики;
- світлофори;
- дорожня розмітка;
- направляючі пристрої.

Пристрої, що забезпечують роботу засобів інформації:

- контролери та мікроконтролери;
- детектори;
- пристрої, які обробляють та передають інформацію;
- засоби зв'язку, в тому числі диспетчерського;
- ЕОМ тощо.

Контролер — це апаратний пристрій або програмне забезпечення, яке керує або спрямовує потік даних між двома об'єктами. В обчислювальній техніці контролерами можуть бути карти, мікросіпи або окремі апаратні пристрої для керування периферійним пристроєм. У загальному сенсі контролер можна розглядати як щось або когось, що взаємодіє між двома системами та керує зв'язком між ними [8, 50].

Дорожні контролери відповідають в свою чергу за перемикання сигналів і символів керованих дорожніх знаків.

Детектори, які використовуються на автомобільному транспорті, є технічними засобами, що контролюють реальну транспортну обстановку на конкретному місці. За допомогою цих детекторів можна забезпечити динамічний контроль руху, стратегічний контроль руху та отримувати необхідну інформацію про дорожній рух.

Транспортні детектори працюють за такими принципами:

- фіксація вібрацій;
- фіксація шумів;
- обробка зображень (телекамери).

Передбачено сім груп дорожніх знаків, які відрізняються між собою формами, кольором та розташуванням, рисунок 1.1.

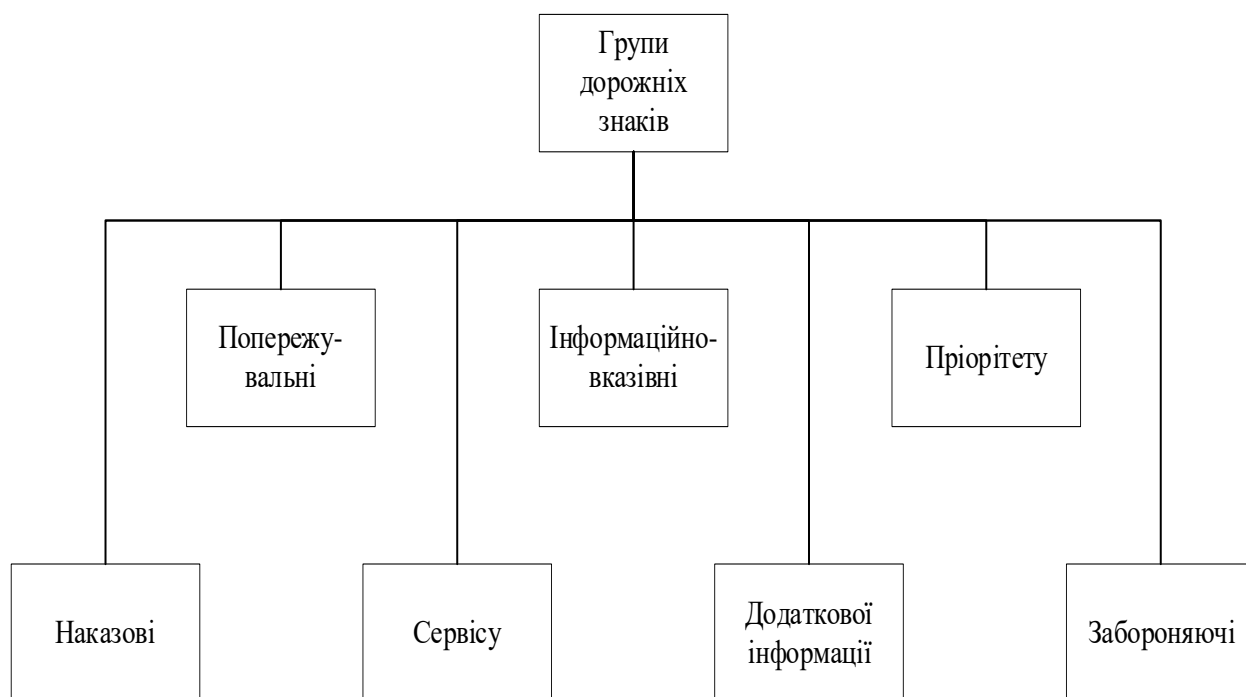


Рисунок 1.1 – Групи дорожніх знаків

Кожен знак або табличка має свою нумерацію. Перша цифра відповідає за номер групи, друга - порядковий номер пластини або таблички в групі. Якщо знак або табличка має різновиди, то в цьому випадку використовується третя цифра - номер різновиду знака [8].

Відповідно до нормативних документів до дорожньої розмітки належать лінії, написи та інші позначки на дорожньому покритті, узбіччях та інших елементах доріг і дорожніх спорудах. Смуга руху зазвичай позначається разом із встановленими дорожніми знаками та світлофорами.

Розглянемо світлофорне регулювання. Саме дане регулювання відносять до технічних засобів організації дорожнього руху.

Світлофор, стоп- сигнали або роботи – це сигнальні пристрої, розташовані на перехрестях доріг, пішохідних переходах та інших місцях для контролю транспортних потоків.

Регулятори сигналів світлофорів – це електронні пристрої, розташовані на перехрестях, які контролюють послідовність увімкнення вогнів. Разом із комп'ютерами, комунікаційним обладнанням і детекторами для підрахунку та вимірювання дорожнього руху контролери часто об'єднуються разом, щоб контролювати велику кількість сигналів світлофора на перехрестях у місті або на з'їздах до швидкісних і автострад. Хоча деталізована марка та тип обладнання значно відрізняються, функції, які виконують системи, загалом узгоджені [8].

Світлофори відповідають універсальному колірному коду, який чергує право проїзду, надане користувачам, з послідовністю освітлювальних ламп або світлодіодів трьох стандартних кольорів:

- зелене світло;
- червоне світло;
- помаранчеве світло.

Зелене світло – дозволяє рухатися в зазначеному напрямку, якщо це безпечно і є місце з іншого боку перехрестя.

Червоне світло – забороняє рух транспорту. Блимаючий червоний індикатор вимагає зупинити рух, а потім продовжити, коли безпечно (еквівалентно знаку зупинки).

Бурштиновий вогник (також відомий як « помаранчеве світло» або « жовте світло») попереджає про те, що сигнал ось-ось зміниться на червоний, при цьому деякі юрисдикції вимагають від водіїв зупинитися, якщо це безпечно, а інші

дозволяють водіям проїхати через перехрестя, якщо це безпечно. У деяких європейських країнах (наприклад, Великобританія) червоний і жовтий відображаються разом, що вказує на те, що сигнал ось-ось зміниться на зелений [11, 31].

Приклад роботи світлофора зображено на рисунку 1.2.

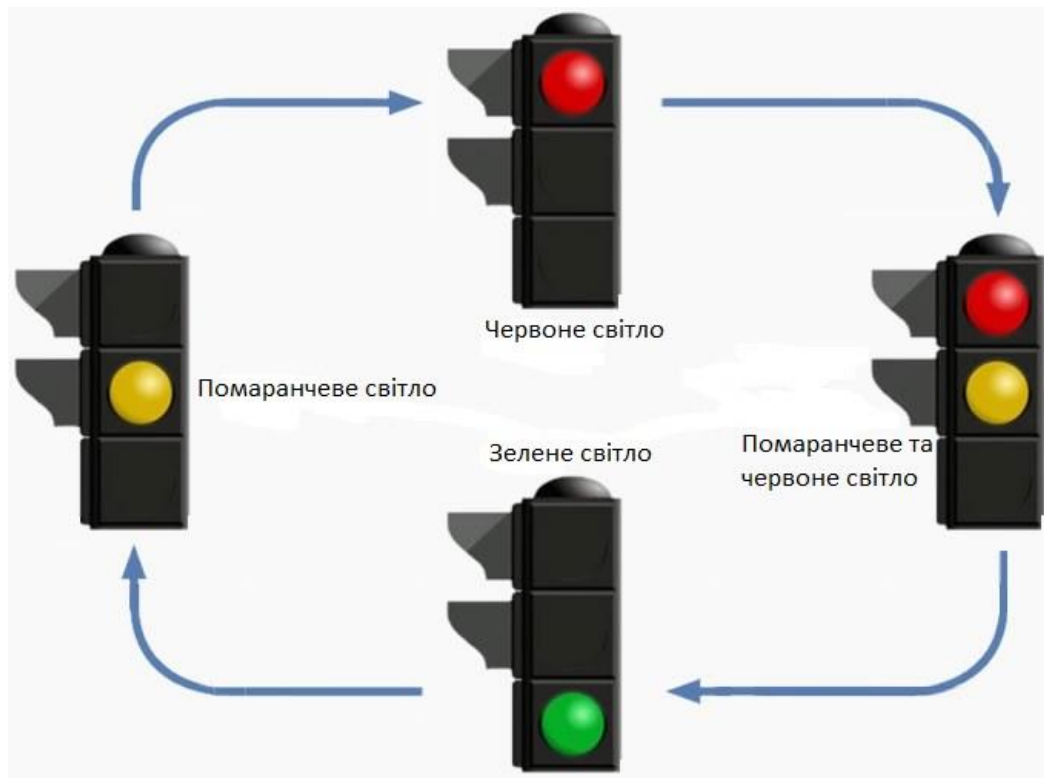


Рисунок 1.2 – приклад роботи світлофора

Світлофори встановлюються на різних ділянках транспортного руху:

- ділянки конфліктуючих транспортних та пішохідних потоків;
- ділянки та смуги, де напрямок руху може змінюватись у протилежний бік;
- на залізничних переїздах, пороммах, мостах (в тому числі розвідних) та переправах;
- на митницях, та прикордонних пунктах;
- на лініях старту спортивних змагань.

Запровадження світлофорного регулювання усуває найнебезпечніші конфліктні точки, що в свою чергу сприяє підвищенню безпеки дорожнього руху.

Водночас встановлення світлофорів на перехресті призводить до затримок руху, тому можливість використання світлофорного регулювання буде досліджено на основі аналізу пов'язаних з цим втрат.

Світлофори мають декілька класифікацій, а саме:

- світлофори, які мають функціональне призначення, які в свою чергу поділяються на транспортні та пішохідні;
- світлофори, які класифікуються за роллю, котру вони виконують в процесі регулювання дорожнім рухом, а саме основні, повторювачі та дублери;
- світлофори з різними видами пересічення.

Запланований вплив на транспортний потік за допомогою світлових сигналів називається світлофорним керуванням. Наприклад, для цього використовується так званий контроль точки розгортання. Це управління світловими сигналами, в якому статуси сигналізації перемикаються через визначені робочі точки. Також можна використовувати контроль сигналу з фіксованим часом. У цьому випадку учасник дорожнього руху не може впливати на керування світлофором (наприклад, через фазу запиту).

Необхідно розрізнити індивідуальний контроль і груповий контроль . При індивідуальному управлінні транспортний потік контролюється за допомогою світлофорної системи без узгодження з іншими світлофорними системами. При груповому управлінні, навпаки, налагоджується координація між кількома світлофорами. Група вузол є вузлом в групі , на якій сигнальні системи трафіку з'єднані один з одним з точки зору технології управління.

Зміну сигналу світлофору в одному контролері забезпечує управління світлофором тільки по одній заздалегідь складеною програмою. Тим часом інтенсивність руху має велике значення змінюється протягом доби і тому програма роботи світлофора (розрахована на якусь певну інтенсивність) не може забезпечити оптимальне регулювання протягом всіх 24 годин. Тобто світлофор працює як таймер та змінює сигнал через деяку одиницю часу циклічно і це головним недоліком.

Саме для вирішення даної проблеми використаємо ультразвуковий давач, який буде вимірювати відстань від пішохідного переходу до автотранспорту, який

наближається. Тобто при наближенні автомобіля до переходу світлофор змінить сигнал для пішоходів з зеленого на червоний, а для водія відповідно навпаки. Це дозволить позбутись заторів, а водії та пішоходи не будуть втрачати дарма свій час [8, 11, 31].

Визначивши основні поняття, було описано типи та принципи роботи дорожніх світлофорів та визначено алгоритм роботи для роботи системи автоматизованого управління світлофорами на основі аналізу руху автотранспорту.

1.2 Аналіз методів автоматизації керування дорожнім рухом

Методи організації дорожнього руху в містах передбачає широке використання комп'ютерних технологій. Сучасний міжнародний досвід показує, що управління рухом у міській вулично-дорожній мережі, насамперед, потребує загальноміської системи управління (інтелектуальної транспортної системи, ІТС).

Розробка моделі соціального процесу є складною задачею, яку можна вирішити за досить жорстких обмежень. Тому більшість завдань управління трафіком не формалізуються, а вирішуються досвідченим шляхом. практичний інтерес і може бути використаний при прогнозуванні мережевих керуючих дій в інтелектуальних транспортних системах, у тому числі в режимі реального часу та для перевантажених ділянок вулично-дорожньої мережі [8, 11, 31, 42].

Для автоматизації перехрестя обладнані системами збору інформації, які включають датчики руху та телевізійні камери. Системами збору інформації фіксує параметри транспортного потоку.

Параметри транспортного потоку зображено на рисунку 1.3.



Рисунок 1.3 – Параметри транспортного потоку

Ця інформація по каналах зв'язку передається в центральний контрольно-розрахунковий комплекс, де аналізується і підбирається програма налаштування світлофора для кожного перехрестя з мінімальною сумарною затримкою для всієї системи, на основі розрахунку ЦКОК формує відповідну команду, яка по лініях зв'язку передається до виконавчих вузлів (контролерів, сервомеханізмів). Зміна системи регулювання призводить до зміни параметрів транспортних потоків.

Надійність системи забезпечується впровадженням систем резервування і використанням комп'ютерів, можливість переходу до жорсткої регламентації кожного світлофорного об'єкта, наявність ручного керування тощо.

Крім того, робота системи візуально контролюється оператором за допомогою мнемосхеми. Мнемосхема являє собою велику карту міста (району), де всі регульовані перехрестя позначені ліхтариками, з'єднаними послідовно з відповідними вузлами ЦКОК. Вони світяться лише тоді, коли світлофори підпорядковані центральному комплексу.

У багатьох країнах швидкісні автомагістралі нещодавно обладнали автоматизованими системами інформування про дорожній рух. Інформація про параметри транспортного потоку, дорожню обстановку, погодні умови надходить до ЦКОК, обробляється в комп'ютері та передається водіям на світлофорах, дорожніх знаках та інформаційних табло зі змінними повідомленнями.

За звичайних умов водіння сигнал не має повідомлень, а поле чисте (зелене). При зміні умов виводяться повідомлення «Обережно! Попереду повільний рух», «Обмеження швидкості», «Потрібно об'їхати ділянку» та інші.

Основна мета цих систем — забезпечити максимальну безпеку шляхом завчасного інформування водіїв про можливі небезпеки та затримки. Таким чином, система має завдання запобігти потраплянню автомобіля в небезпечну ситуацію.

Розглянемо методи регулювання світлофорами. Основними елементами світлофорної системи є регулювач, світлофорні головки та детектор. Діючи як «мозок» за функціонуванням системи освітлення, контролер містить необхідну інформацію для спрямування світла через різні послідовності залежно від часу доби та розташування системи. Детектори використовуються для виявлення виїзду автомобілів на перехрестя, виїзду автомобілів на поворотні смуги (щоб увімкнулися стрілки) або скупчення великої кількості автомобілів на перехрестях.

Існує два основних типи роботи сигналів світлофора: фіксований період і на основі сигналу під'їзду транспортного засобу. Сигнали світлофора з фіксованим періодом змінюються відповідно до попередньо встановлених часових проміжків (інтервалів) і мають циклічні сигнали постійно, навіть якщо немає пішоходів або транспортних засобів. Світлофори з передбачуваною схемою руху добре працюють на дорогах із передбачуваною схемою руху та низькою швидкістю.

На перехрестях доріг із різною схемою руху та високими швидкостями необхідні світлофори, які можуть виявляти присутність транспортних засобів. Насправді світлофори на перехрестях мають розпізнавати не лише вантажівки, автомобілі та автобуси, а й велосипеди, мотоцикли та пішоходів.

Деякі світлофори змінюють сигнали лише тоді, коли на бічних вулицях є транспортні засоби або пішоходи. Їхній сигнал на вулицях змінюється залежно від заторів, і є запрограмовані на максимальне та мінімальне значення.

Швидкий і постійний прогрес в комунікаційних та комп'ютерних технологіях стимулюють безліч нових концепцій в управлінні дорожнім рухом. Автомобілі, оснащені бортовими комп'ютерами, дисплеями водія та комунікаційними пристроями, отримують інструкції щодо оптимального шляху до пункту призначення від центру управління дорожнім рухом. Транспортний засіб також періодично повідомлятиме про свій час у дорозі та швидкість, який використовуватимуться як частина інформації для комп'ютера для надання порад. У більш просунутих системах час сигналів світлофора на перехрестях і з'їздах узгоджуватиметься з порадами про маршрут. Замість того, щоб просто пристосовуватись до транспортних засобів, які подорожують мережею, система призведе до зміни моделей подорожей. Комп'ютери та датчики в автомобілі контролюватимуть роботу критичних систем безпеки (наприклад гальма, рульове керування), попереджаючи водія, коли умови перевищують номінальні значення.

Зв'язок і комп'ютери також допоможуть рухатися вантажівкам та іншому комерційному транспорту у містах. Диспетчер зможе змінити розклад, поки водій у дорозі. Для цих компаній це означає зниження витрат, а для їхніх клієнтів покращення обслуговування. Водіїв у далеких міжміських поїздах можна попередити про наближення негоди. Вони також можуть отримувати попередження, якщо надто швидко в'їжджають на поворот або на перехрестя. Завдяки таким системам безпека на дорозі повинна значно підвищитися.

Користувачі громадського транспорту повинні мати можливість отримувати точнішу інформацію про час у дорозі та наявність місць в автобусах і поїздах. Якщо точну інформацію можна надати вдома чи в офісі, системи можуть розподілити пікове навантаження, що зробить обслуговування дешевшим, а поїздки комфортнішою для мандрівника. Ті, хто поінформований про затори або некомфортні умови, можуть скористатися іншою системою, щоб знайти пару для учасника парку автомобілів. Крім того, люди можуть «дистанційно працювати», залишаючись вдома та працюючи в офісі в електронному вигляді.

Нарешті, остаточна система розглядається як автоматична система керування транспортним засобом, у якій транспортний засіб водія перевіряється на авторизованій станції, а потім їде по шосе, смузі чи місцевій вулиці. Відстань до транспортного засобу, що їде попереду, і бокове керування в межах смуги визначають бортові комп'ютери. Очікується, що максимальний потік зросте з 2000 транспортних засобів на годину на смугу до 10-20 тисяч. Збільшення потоків означатиме суттєве зменшення заторів і, оскільки транспортні засоби автоматично контролюються, покращується безпека на дорозі, що призведе до зменшення аварій через помилку водія [8, 11, 31, 42].

1.3 Аналіз засобів реалізації системи управління світлофорами

Мікроконтролер — це одна інтегральна схема, яка складається з різних елементів, включаючи мікропроцесор, таймери, лічильники, порти вводу/виводу (I/O), оперативну пам'ять (RAM), постійну пам'ять (ROM) і деякі інші компоненти. Ці частини працюють разом для виконання задалегідь запрограмованого набору конкретних завдань. Таким чином, мікроконтролер схожий на маленький комп'ютер, який обробляє і навіть виконує керування в електронному пристрої. Мікроконтролери призначені для вбудованих додатків, на відміну від мікропроцесорів, що використовуються в персональних комп'ютерах або інших додатках загального призначення, що складаються з різних дискретних мікросхем [20, 49, 50].

Слід зазначити, що досить часто неправильно позначають плати розробки, такі як розроблені Arduino, як мікроконтролери. Однак Arduino точніше описувати як платформу розробки, яка зосереджена навколо легко програмованих плат із мікроконтролерами в основі. Сама «Arduino» — це просто складна плата, на якій розміщено всі компоненти мікроконтролера, а також деякі додаткові порти та інші функції. Це дозволяє програмувати та перепрограмувати мікроконтролер для різних завдань.

Збір даних – це процес вибірки сигналів, які вимірюють реальні фізичні умови та перетворення отриманих вибірок у цифрові числові значення, якими можна керувати за допомогою комп’ютера. Системи збору даних, зазвичай перетворюють аналогові сигнали в цифрові значення для обробки. До складу систем збору даних входять:

- давачі для перетворення фізичних параметрів в електричні сигнали;
- схема формування сигналу для перетворення сигналів давачів у форму, яку можна перетворити на цифрові значення;
- аналого-цифрові перетворювачі для перетворення умовних сигналів датчиків у цифрові значення.

Системи збору даних зазвичай керуються програмами, розробленими з використанням різних мов програмування загального призначення, таких як Assembly , BASIC , C , C++ , C# , Fortran , Java , LabVIEW , Lisp , Pascal тощо.

Для реалізації системи автоматизованого управління світлофорами в першу чергу обрати відповідний мікроконтролер. Проаналізувавши ринок мікроконтролерів, було обрано мікроконтролер Arduino, оскільки він має ряд плюсів:

- простота у використанні;
- низька ціна;
- відкритий вихідний код;
- безкоштовне середовище для розробки;
- надійність.

Arduino Mega 2560 — це плата мікроконтролера на основі ATmega2560 . Він має 54 контакти цифрового входу/виходу (з яких 15 можна використовувати як виходи ШІМ), 16 аналогових входів, 4 UART (апаратні послідовні порти), кварцевий генератор 16 МГц, з’єднання USB, роз’єм живлення, роз’єм ICSP, і кнопку скидання. Містить все необхідне для підтримки мікроконтролера [20, 49, 50].

Ультразвуковий давач відстані HC-SR04 – це давач, який використовується для визначення відстані до об’єкта за допомогою сонара. Він ідеально підходить для будь-яких проектів з робототехніки, які вимагають уникати об’єктів, виявляючи, наскільки близько вони знаходяться.

Розглянемо структурну схему роботи мікроконтролера. Структурна схема зображена на рисунку 1.4.



Рисунок 1.4 – Структурна схема збору аналогової інформації

З структурної схеми можна зрозуміти, що до входів мікроконтролера будуть під’єднані давачі, а на вихід – пристрої виводу. Тобто роль давачів – збирати інформацію і подавати її на входи мікроконтролера. Роль мікроконтролера – обробляти інформацію та видавати її на входи виходу, пристроями виводу у свою чергу являється світлофором.

Враховуючи вище наведену структурну схему та обрані засоби реалізації проекту створимо синтез структур системи управління світлофорами на основі аналізу руху автотранспорту з використанням І-АБО дерева.

Синтез множини структур з використанням І-АБО дерева зображено на рисунку 1.5.

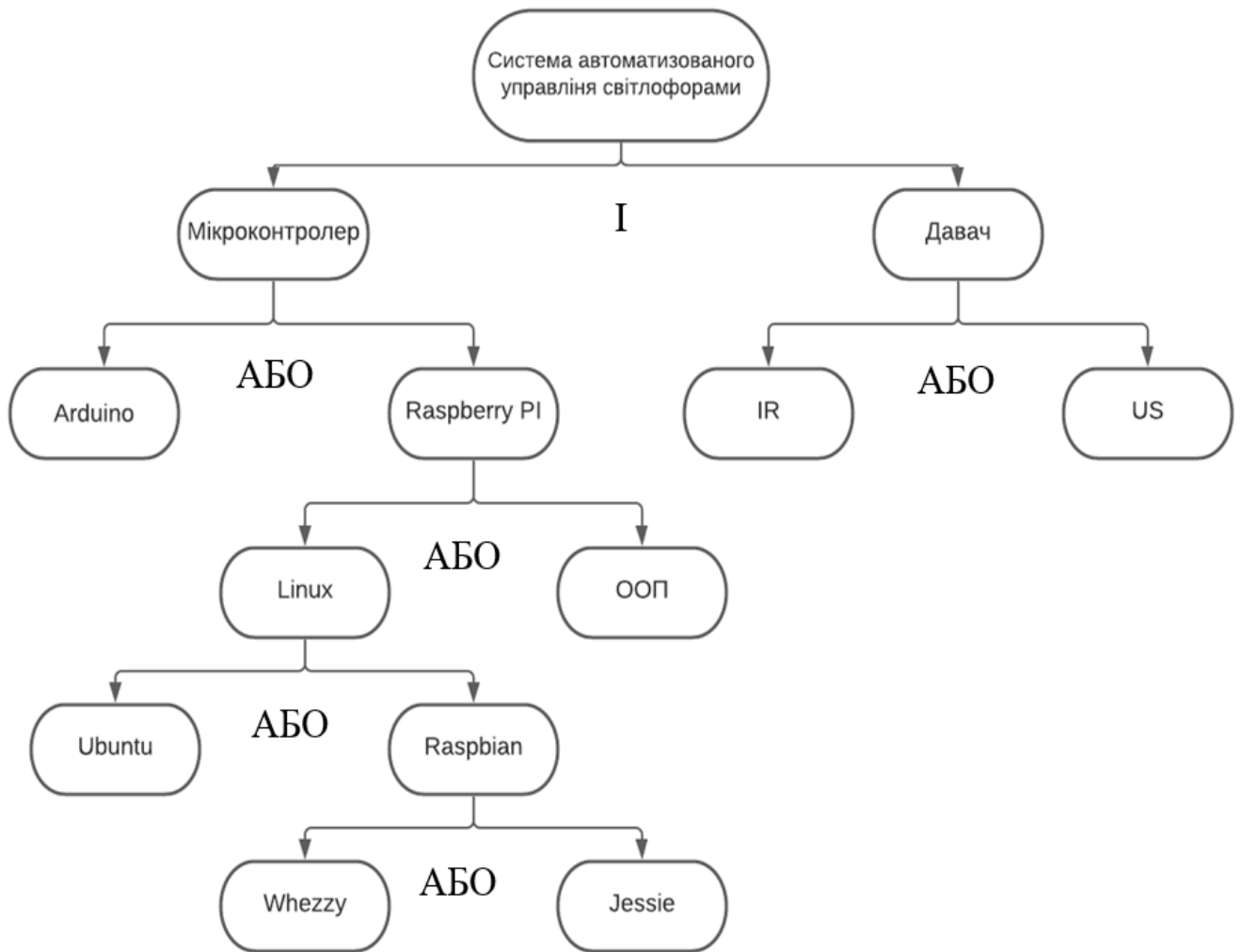


Рисунок 1.5 – Синтез множини структур з використанням І-АБО дерева

Даний синтез множини структур з використанням І-АБО дерева системи управління світлофорами на основі аналізу руху автотранспорту вказує нам на альтернативні засоби досягнення мети.

Враховуючи дані синтез множини структур системи управління світлофорами на основі аналізу руху автотранспорту знайдемо оптимальний варіант структури з адитивного критерію.

Пошук оптимального варіанту структури з адитивного критерію зображений на рисунках 1.6, 1.7, 1.8.

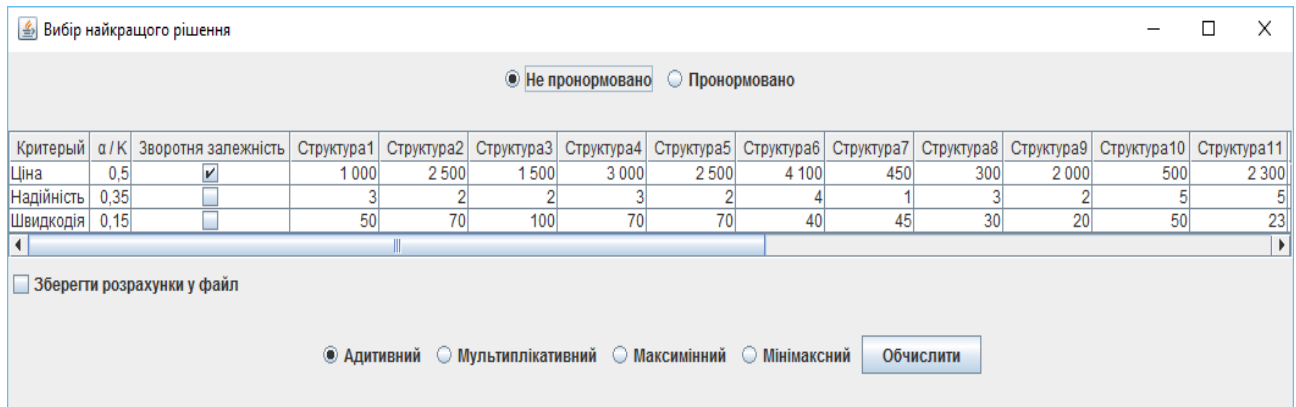


Рисунок 1.6 – Пошук оптимального варіанту структури з адитивного критерію

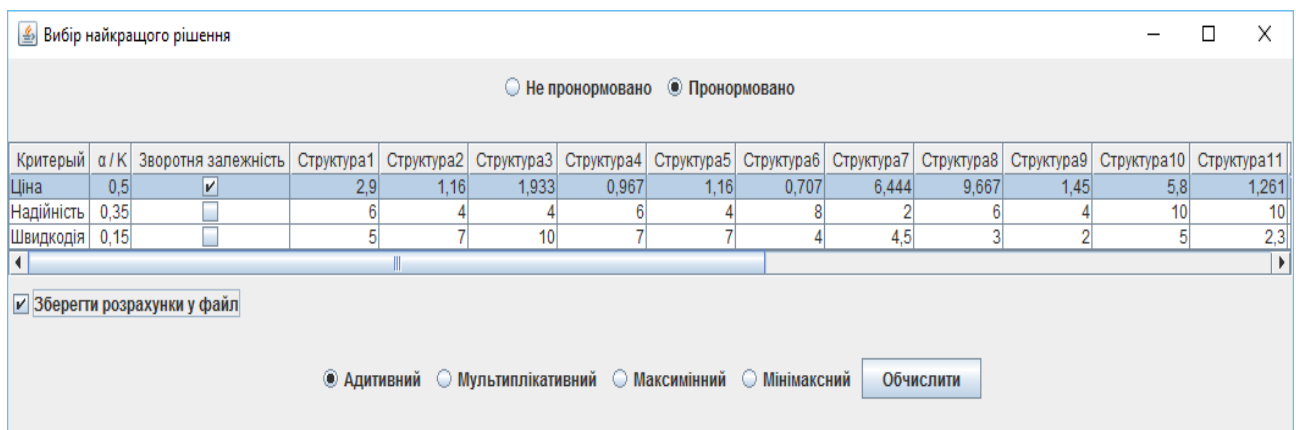


Рисунок 1.7 – Пошук оптимального варіанту структури з адитивного критерію

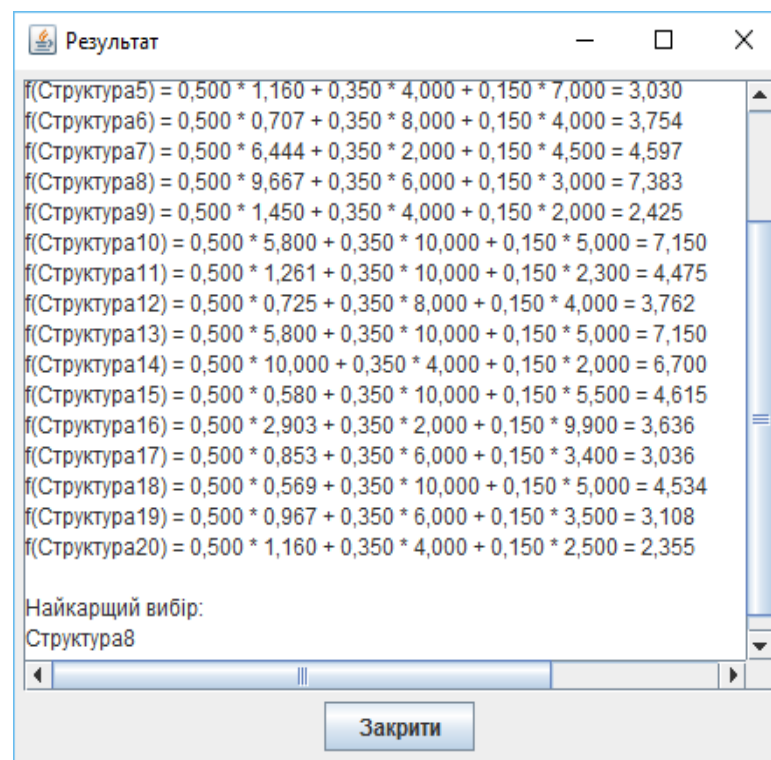


Рисунок 1.8 – Пошук оптимального варіанту структури з адитивного критерію

На основі даних пошук оптимального варіанту структури з адитивного критерію ми отримали найбільш оптимізовану структуру системи управління світлофорами на основі аналізу руху автотранспорту.

Вище обрані засоби, а саме взятий за основу мікроконтролер Arduino і точний давач відстані HC-SR04 як і було розраховано найбільше підходять для розробки системи автоматизованого управління світлофорами на основі аналізу руху автотранспорту, за рахунок надійності системи, собівартості та швидкодії обраних елементів.

Для оптимізації системи управління світлофорами потрібно не лише проаналізувати технічні засоби, а підійти до питання комплексно і зрозуміти на якій ділянці система дійсно потрібна. Для даної використаємо технологію комп'ютерного зору.

Комп'ютерний зір — це сфера штучного інтелекту (ШІ), яка дозволяє комп'ютерам і системам отримувати значущу інформацію з цифрових зображень, відео та інших візуальних вхідних даних, а також виконувати дії або давати рекомендації на основі цієї інформації. Якщо ШІ дозволяє комп'ютерам мислити, комп'ютерний зір дозволяє їм бачити, спостерігати та розуміти [1].

Комп'ютерний зір працює подібно до людського, але люди мають перевагу. Людський зір тренувався протягом багатьох років, щоб розрізняти об'єкти, оцінювати відстані, виявляти рух і помічати нерівності в зображенні.

Комп'ютерний зір навчає машини цим навичкам, але робить це за набагато менший час, використовуючи камери, дані та алгоритми замість сітківки, зорових нервів і зорової кори. Система, навчена для перевірки продукту або моніторингу виробничої лінії, може перевіряти тисячі продуктів або процесів за хвилину та виявляти недоліки або проблеми, які ледь помітні, швидко перевищуючи людські можливості.

Комп'ютерне бачення використовується в різних галузях промисловості, від енергетики та комунальних послуг до виробництва та автомобілебудування, і ринок продовжує зростати.

Комп'ютерний зір вимагає великих обсягів даних. Дані багаторазово аналізуються, доки не можна виявити відмінності та, зрештою, розпізнати

зображення. Наприклад, якщо комп'ютер потрібно навчити розпізнавати автомобільні шини, йому потрібно завантажувати величезну кількість зображень шин і пов'язаних із ними елементів, щоб дізнатися про відмінності та розпізнати шину, особливо без дефектів.

Використовуються дві основні технології: тип машинного навчання під назвою глибоке навчання та так звана нейронна мережа.

Машинне навчання використовує алгоритмічні моделі, які дозволяють комп'ютеру інформувати себе про контекст візуальних даних. Коли через модель буде подано достатньо даних, комп'ютер «перегляне» дані та навчиться відрізняти одне зображення від іншого. Алгоритми дозволяють машині навчатися самостійно, на відміну від програмування людиною.

Нейронна мережа допомагає моделі машинного або глибокого навчання «переглядати» зображення, розбиваючи їх на пікселі, позначені тегами або мітками. Мітки використовуються для виконання згорток (операція, коли дві функції додають до третьої) і прогнозування того, що буде «побачено». Нейронна мережа виконує згортки та перевіряє точність своїх передбачень протягом кількох ітерацій, доки прогнози не почнуть щоб збутися, він розпізнає або бачить зображення, схожі на людей.

Подібно до того, як людина розпізнає зображення на відстані, нейронна мережа спочатку розпізнає жорсткі краї та прості форми, а потім додає інформацію, повторюючи свої передбачення. Нейронна мережа використовується для розуміння зображень по одному. Повторювана нейронна мережа так само використовується у відео-додатках, щоб допомогти комп'ютерам зрозуміти зв'язок між зображеннями в серії кадрів.

У загальному можна виділити шість основних методів (алгоритмів) комп'ютерного бачення, які використовують у різних сферах, включаючи і автотранспортну та дорожню сфери а саме:

- класифікація зображень;
- виявлення об'єктів;
- відстеження об'єктів;
- семантична сегментація;

- сегментація екземпляра;
- реконструкція зображень.

Класифікація зображень найвідоміша техніка комп'ютерного зору. Однією з найбільших проблем, яку потрібно подолати у даній технології, є зміна масштабу, точки зору, деформація зображення, умови освітлення.

Для створення алгоритмів комп'ютерного зору, які зможуть класифікувати зображення за відповідними категоріями існує дуже цікавий підхід, керований даними. Замість того, щоб визначати, як буде виглядати кожна категорія зображень на рівні коду, дослідник дає комп'ютеру багато прикладів класу зображень для машинного навчання з комп'ютерним зором. Комп'ютер повинен вивчити зображення та дізнатись про їх візуальний вигляд [3].

Виявлення об'єктів - це техніка комп'ютерного зору, яка працює для ідентифікації та виявлення розташування об'єктів у зображенні чи відео. Зокрема, виявлення об'єктів має обмежувальні рамки навколо них, які дозволяють нам зрозуміти, де ці об'єкти знаходяться (або як вони рухаються) через дану сцену.

Розпізнавання об'єктів зазвичай плутають із розпізнаванням зображень, тому перед тим, як продовжити, важливо з'ясувати відмінності між ними.

Розпізнавання зображень присвоює зображенню мітку. Модель передбачає, де знаходиться кожен об'єкт і яку мітку слід застосовувати. Таким чином, виявлення об'єктів надає більше інформації про зображення, ніж розпізнавання.

Предмет виявлення може бути розбите на машинне навчання на основі підходів глибокого навчання. У більш традиційних підходах, заснованих на машинному навчанні, методи комп'ютерного зору використовуються для розгляду різних особливостей зображення, таких як кольорова гістограма або краї, для визначення груп пікселів, які можуть належати об'єкту. Потім ці ознаки подаються в модель регресії, яка передбачає розташування об'єкта разом з його міткою [7].

Відстеження об'єктів – відслідковування одного або декількох рухомих об'єктів у будь-якій заданій сцені. Це традиційно застосовувалося для моніторингу взаємодій у реальному світі після виявлення вихідного об'єкта. Це

дуже важливий компонент самокерованих автомобілів, який планують випустити такі компанії, як Uber і Tesla. Відстеження об'єктів можна розділити на дві категорії: генеративне і дискримінаційне. Генеративний метод описує очевидні характеристики і зменшує ймовірність на помилку в реконструкції при пошуку об'єкта.

Дискримінаційний підхід є більш потужним і точним. Він може бути використаний для визначення різниці між об'єктом і фоном, ставши кращим методом відстеження. Він також називається відстеженням по виявленню, який відноситься до тієї ж категорії, що і глибоке навчання [3].

Сегментація зображень – це процес розподілу зображення на кілька сегментів. У цьому процесі кожен піксель на зображенні асоціюється з типом об'єкта. Існує два основних типи сегментації зображень – семантична та сегментація екземплярів.

У семантичній сегментації всі об'єкти одного типу позначаються за допомогою однієї мітки класу, тоді як, наприклад, подібні об'єкти сегментації отримують свої окремі мітки. Основна архітектура сегментації зображень складається з кодера та декодера. Кодер витягує функції із зображення за допомогою фільтрів. Декодер відповідає за генерування кінцевого виводу, який зазвичай є маскою сегментації, що містить контур об'єкта.

Сегментація екземпляра – це уточнена версія семантичної сегментації. Такі категорії, як "транспортні засоби", поділяються на "машини", "мотоцикли", "автобуси", "вантажівки", тощо – сегментація екземплярів виявляє випадки кожної категорії [6].

Іншими словами, семантична сегментація трактує кілька об'єктів в межах однієї категорії як одну сутність. Натомість сегментація екземплярів визначає окремі об'єкти в межах цих категорій [6].

Реконструкція зображень полягає у відновленні зображення. Набори даних, як правило, включають поточні набори фотографій, щоб знайти пошкоджені версії зображення, які моделі повинні навчитися ремонтувати [3].

Враховуючи вище наведений аналіз комп'ютерного зору, можемо зробити висновок, що дану технологію можна використати для аналізу руху автотранспорту, а саме інтенсивності руху, та підрахунку загальної кількості автотранспорту на окремій ділянці дороги.

1.4 Постановка задачі кваліфікаційної роботи

Сьогодні люди проводять більшу частину свого часу поза домом, вони щодня їздять на роботу, часто відвідують торгові центри та пам'ятки, не забуваючи про переміщення в центр міста. Це, безсумнівно, спричинило дисбаланс у повсякденній мобільності. Це, з одного боку, спричиняє додаткові викиди вуглекислого газу та завдає шкоди навколишньому середовищу екосистеми міста. З іншого боку, це збільшує розчарування водія та затори в місті, що неодмінно спричинить дорожньо-транспортні пригоди.

Аналіз методів автоматизації керування дорожнім рухом показав, що більшість із них характеризується суттєвими недоліками, зокрема – вимагають великих вибірок даних, низькою точністю, використовують складні обчислювальні процедури, вимагають значних часових затрат на побудову моделей.

Для задачі ідентифікації автоматизації управління транспортними потоками важливою є висока швидкість виявлення автотранспорту з метою зменшення ризику дорожніх транспортних пригод пошкодження та часу проведення водієм у дорозі. У такому випадку ефективним є застосування ультразвукового давача відстані та методів комп'ютерного зору.

Зважаючи на це, у кваліфікаційній роботі слід виконати такі завдання:

- провести аналіз засобів регулювання дорожнього руху;
- охарактеризувати методи автоматизації керування дорожнім рухом;
- розробити алгоритм оптимізації транспортних потоків на автострадах;
- удосконалити схему оптимізації транспортних потоків на

автомагістралях;

- розробити програмно-апаратний комплекс для реалізації алгоритму оптимізації транспортних потоків на автострадах;
- провести експериментальні дослідження ефективності розробленого алгоритму оптимізації транспортних потоків на автострадах.

1.5 Висновки до розділу

Проаналізовано основні засоби реалізації системи управління транспортними потоками описано основні методи їх класифікації. Зазначено, що основним технічним пристроєм автоматизації дорожнього руху є світлофор. Проведено аналіз засобів реалізації системи управління світлофорами. Досліджено існуючі автоматизовані системи управління транспортними потоками, виокремлено основні їх недоліки. Здійснено постановку задачі на кваліфікаційну роботу на основі проведеного аналізу.

2 АЛГОРИТМ АВТОМАТИЗОВАНОГО РЕГУЛЮВАННЯ ДОРОЖНЬОГО РУХУ

2.1 Системи керування дорожнім рухом

Кілька систем керування дорожнім рухом були впроваджені в останні роки з використанням різних технологій зв'язку та спостереження для контролю та управління проблемою міського руху в містах і вирішення обмежень традиційних систем світлофорів [22, 28, 32, 47].

Р. Зітуні, Дж. Петі, А. Джуді та Л. Джордж у своїй роботі запропонувати нову архітектуру для системи управління міським рухом на основі мережі IoT. Ця система дає змогу підключати дороги до Інтернету через сенсорні вузли, здатні виявляти прибуття транспортних засобів і надсилати виявлені дані в хмару з прикордонного маршрутизатора. Дані, зібрані в хмарі, дозволяють проміжному ПЗ визначити майбутній стан світлофорів. Це рішення поширюється через мережу на виконавчі пристрої, встановлені на світлофорах для керування рухом на міських перехрестях. Ця система заснована на реалізації протоколу самоорганізації, який створює топологію мережі зі зіркою, що дозволяє всім вузлам виявлення надсилати свої дані до вузла-приймача через один стрибок [22, 34].

Однак, цей протокол має недоліки, оскільки він створить дисбаланс навантаження між різними вузлами виявлення під час передачі даних одного стрибка до вузла-приймача, швидко виснажуючи найвіддаленіші вузли, що негативно вплине на якість зв'язку та надійність системи. Крім того, цей підхід використовує останні технології, такі як бездротові датчики, щоб обмежити вартість розгортання системи. Однак таке рішення залишається застарілим щодо громадян і водіїв, оскільки вони не можуть підключитися до доріг і знати стан дорожнього руху в режимі реального часу та віддалено, що є частиною концепції створення розумних міст. Структура системи управління міським рухом на основі мережі IoT зображена на рисунку 2.1.

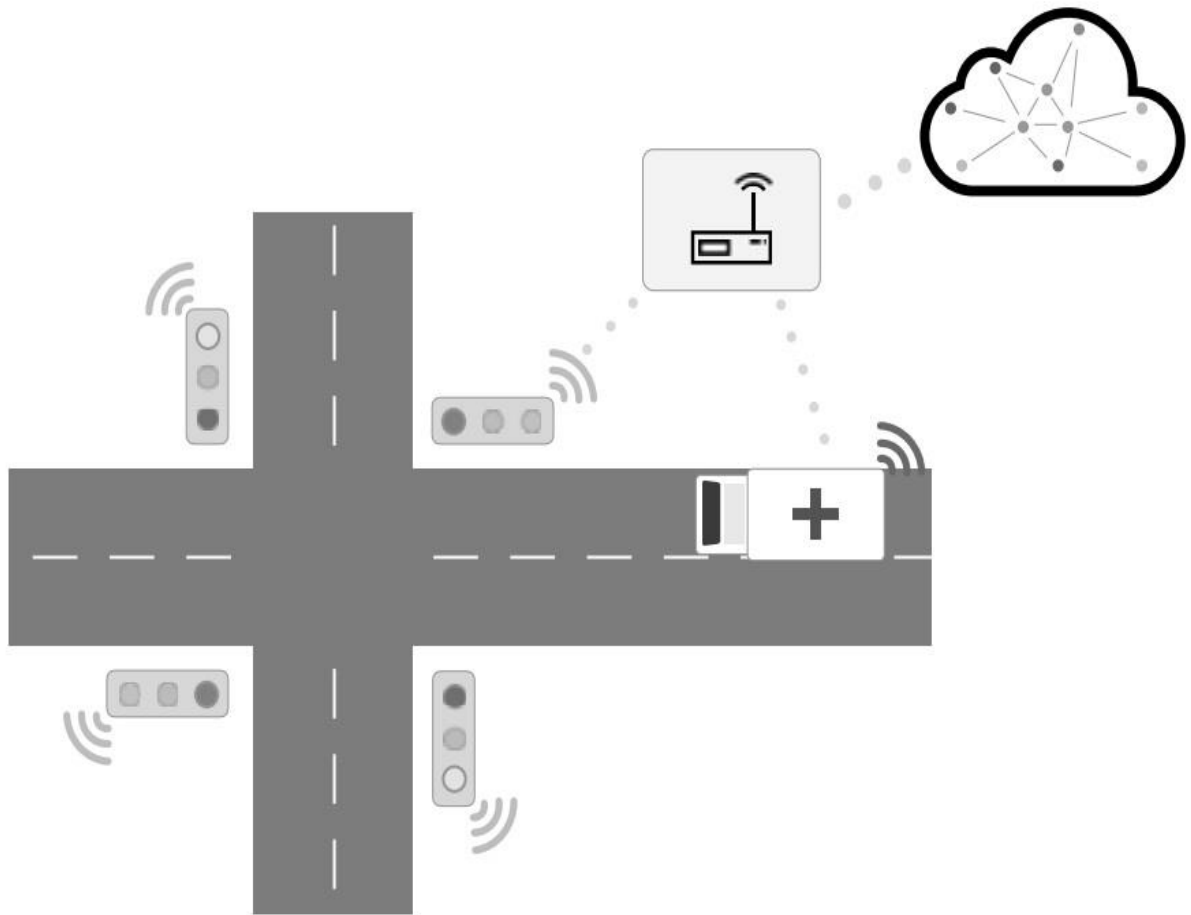


Рисунок 2.1 – Система управління міським рухом на основі мережі IoT

У роботі С. Латіф, Х. Афзаал і Н. А. Зафар, представлена нова інтелектуальна система керування дорожнім рухом, яка базується на розгортанні бездротових сенсорних мереж на дорогах, світлофорах і в певних місцях (таких як лікарні та бензоколонки) для моніторингу дорожнього руху в міста та знайти найкоротший за часом і відстанню шлях до місця призначення, уникаючи заторів. Ця система використовує інтелектуальні камери на дорогах, щоб ідентифікувати номери транспортних засобів і надсилати цю інформацію в центральну систему для моніторингу автомобілів у місті. Запропонована система використовує новітні технології, які дозволяють об'єднувати між собою різноманітні міські служби шляхом створення розумного міста. Однак розгортання інтелектуальних камер може бути дорогим і менш ефективним [22, 33].

П. Садхухан і Ф. Газі пропонують інтелектуальний контроль заторів, заснований на розгортанні бездротових сенсорних мереж для вимірювання

щільності дорожніх заторів, створених на перехрестях. Ця система складається з двох модулів. Перший — це TDMM (модуль моніторингу щільності руху), який використовує ультразвуковий датчик для вимірювання довжини черги, створеної натопом автомобілів, а другий — TMM (модуль керування дорожнім рухом), який є програмним забезпеченням, розгорнутим у комп'ютері, що робить можливим керувати світлофорами відповідно до даних, зібраних різними TDMM, встановленими на дорогах. TDMM, який розгорнуто на кожній дорозі на перехресті, надсилає зібрані дані до найближчого TMM через Wi-Fi, використовуючи з'єднання з кількома або одним проміжком зв'язку залежно від діапазону зв'язку, щоб визначити щільність заторів на дорозі (сильний, середній або низький) і динамічно визначають час роботи світлофора відповідно до значень, отриманих з різних маршрутів. Ця система використовує протокол самоорганізації, який створює неавтономну топологію типу дерева між різними вузлами, з яких кожен вузол моніторингу спілкується через один стрибок із найближчим вузлом, який, у свою чергу, спілкується з вузлом-приймачем через проміжні вузли для передачі даних до модуля керування трафіком. Однак неавтономна деревоподібна структура, сформована цією системою, створює дисбаланс у споживанні енергії між різними вузлами моніторингу, особливо для проміжних вузлів, а також знижує якість доставки даних до центрального вузла, коли один із вузлів маршрутизації стає несправний або виснажений енергією. На додачу, розгортання ультразвукових датчиків слід використовувати лише на дорогах з невеликим і помірним трафіком. Виявлення автомобіля на кількох смугах за допомогою придорожніх ультразвукових датчиків знижує точність виявлення в інтенсивному русі. Це може спричинити проблеми зі стабільністю системи, особливо в години пік із інтенсивним трафіком, що може призвести до неправильних рішень під час оцінювання часу роботи світлофорів [17, 22].

Система інтелектуального контролю заторів, заснованою на розгортанні бездротових сенсорних мереж, зображено на рисунку 2.2.

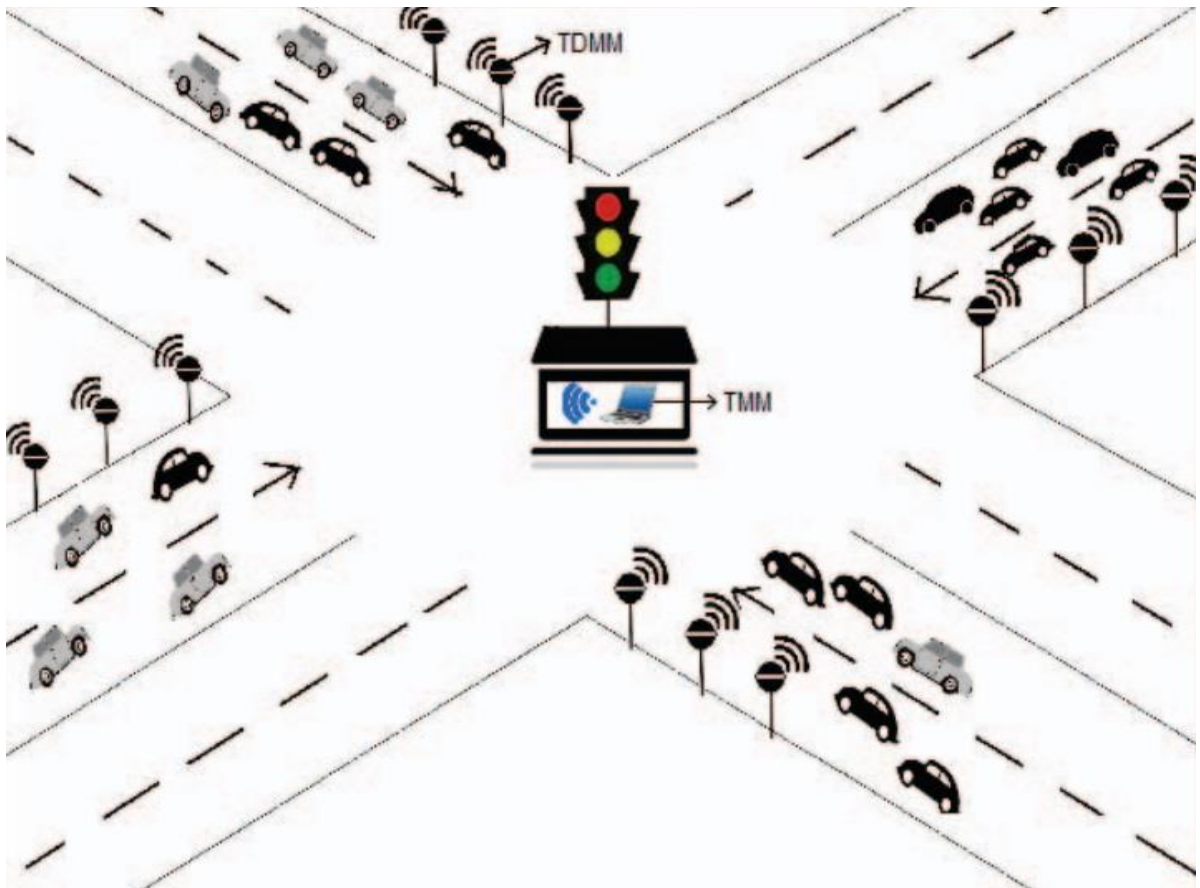


Рисунок 2.2 – Система інтелектуального контролю заторів, заснованого на розгортанні бездротових сенсорних мереж

Автори М. Левандовські, Б. Плачек, М. Бернас і П. Шимала пропонують систему моніторингу дорожнього руху на основі мобільних пристроїв і Bluetooth-маяків з низьким енергоспоживанням. Виявлення транспортних засобів, запропоноване цією системою, використовує мобільні пристрої (наприклад, смартфони), встановлені на узбіччі дороги, для вимірювання потужності сигналу RSSI під час отримання радіочастотних кадрів, випромінюваних маяками Bluetooth на іншій стороні вулиці. Bluetooth-маячки встановлюються уздовж дороги на різній висоті, щоб ідентифікувати та класифікувати тип транспортних засобів, що рухаються по дорозі (легкові або вантажні). Значення RSSI, визначені мобільними пристроями на кожному маршруті, а також їх положення надсилаються через стільникову мережу або зв'язок Wi-Fi на сервер для вимірювання щільності дорожніх заторів і моніторингу трафіку на дорогах, рисунок 2.3.

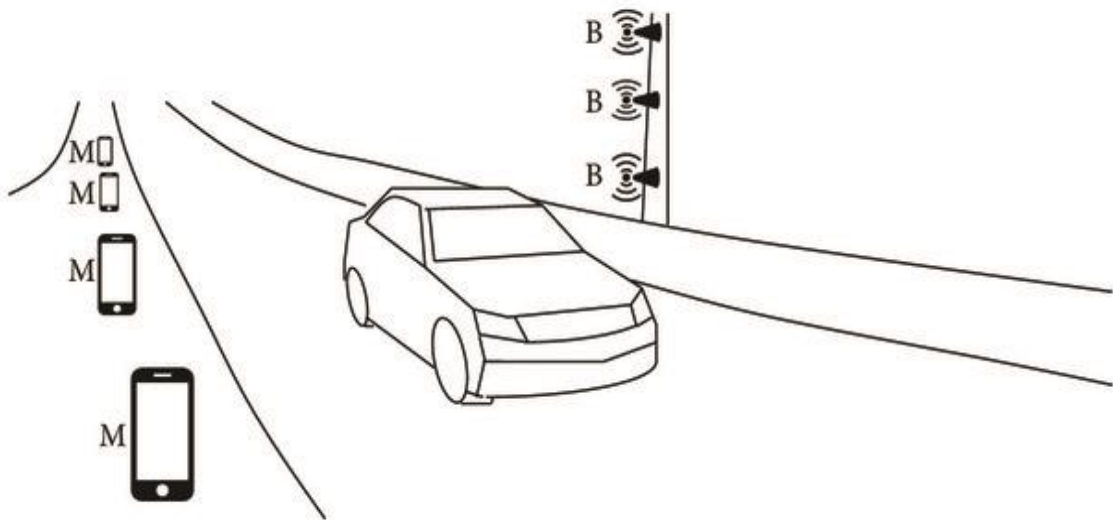


Рисунок 2.3 – Система моніторингу дорожнього руху на основі мобільних пристроїв і Bluetooth-маяків з низьким енергоспоживанням

З іншої сторони, Технологія Bluetooth може спричинити серйозні проблеми з синхронізацією та порушення зв'язку між маяком BLE і смартфоном, що негативно впливає на працездатність системи, особливо у випадку інтенсивного трафіку. Отже, агент має бути на місці, щоб з'єднати два пристрої, щоб відновити зв'язок [22, 38].

Автори в В. Чжан, Г. Тан, Н. Дін і Г. Ван, презентують нову інтелектуальну систему моніторингу дорожнього руху та керування світлофорами на основі бездротових сенсорних мереж. Ці сенсорні вузли встановлюються вздовж доріг, які є перехрестям. Дані, отримані датчиками, надсилаються до контролера з двома світлофорами, щоб оцінити умови заторів на кожній дорозі на перехресті та спрогнозувати стан заторів. Ця система використовує протокол самоорганізації (Alg5), який створює зіркоподібну топологію між різними вузлами мережі. Однак алгоритм, прийнятий цією системою, створить темні області для певних вузлів, розташованих далеко від пов'язаного з ними центрального вузла, з якими вони не зможуть спілкуватися з ним, і це призведе до погіршення якості та здійсненості цієї системи. Це рішення дозволяє динамічно керувати світлофорами відповідно до отриманих станів заторів на перехресті, а також дає можливість оптимізувати фазу синхронізації управління світлофорами, щоб уникнути заторів до їх

утворення. Інтелект цієї системи залишається поза досяжністю водіїв і громадян, оскільки вони не взаємодіють з дистанційною системою, а також не підключаються до доріг у режимі реального часу [22, 41].

Автори в К. Заторі, М. Єреді та Т. Езедіне, описують нову інтелектуальну систему адаптивного керування світлофорами, засновану на розгортанні бездротової сенсорної мережі (WSN) на проїжджій частині, що веде до перехрестя. Ці вузли являють собою магнітні датчики, встановлені в землі вздовж усіх шляхів, які утворюють перехрестя. Ці датчики утворюють топологію мережі кластерного типу, в якій кожен вузол виявляє присутність транспортних засобів і надсилає дані найближчому головному кластеру для досягнення базової станції. Дані, зібрані WSN, використовуються базовою станцією шляхом запуску алгоритму для визначення рівня заторів на кожній смузі та динамічного керування світлофорами на перехресті. Передача даних на базову станцію сенсорними вузлами базується на використанні само-організаційного протоколу, який дозволяє всім вузлам мережі формувати кластерну топологію, головки кластера (CH) передають усі дані свого кластера на базової станції за один стрибок [15, 22].

Однак цей протокол створює дисбаланс навантаження між деякими вузлами датчиків, які обрані головними кластерами та які знаходяться далеко від базової станції, оскільки зв'язок з базовою станцією за один стрибок споживає багато енергії, що спричиняє швидке виснаження цих вузлів. У цій системі використовується новітня інтелектуальна та недорога технологія для моніторингу заторів і керування світлофорами. Підхід до виявлення транспортного засобу зображено на рисунку 2.4.

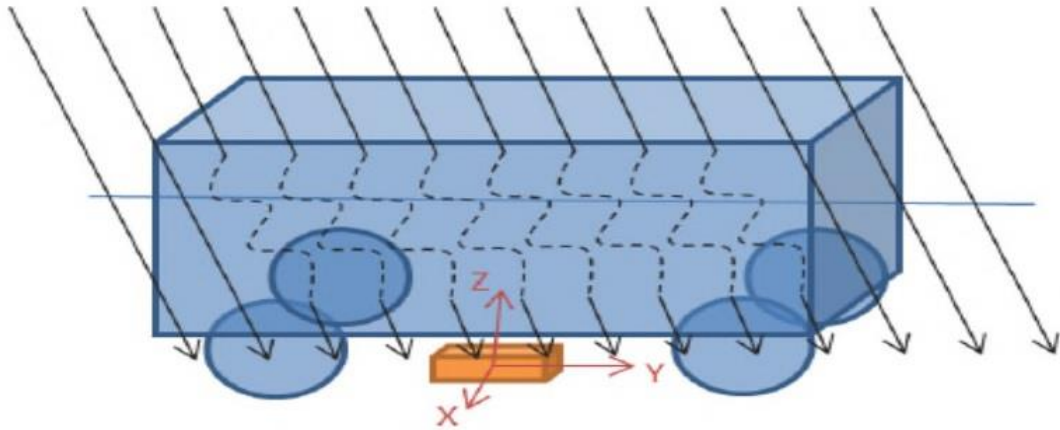


Рисунок 2.4 – Підхід до виявлення транспортного засобу

Беручи до уваги проведений аналіз алгоритмів оптимізації транспортних потоків, розробимо систему оптимізації транспортних потоків.

2.2 Методи оптимізації регулювання дорожнім рухом

У централізованих системах управління трафіком реалізовані методи оптимізації отриманих сигналів світлофора з відповідними вимогами до руху. Існує багато методів оптимізації, і вибір правильного методу повинен враховувати проблему керування, час обчислення та будь-які інші фактори.

Для пошуку оптимального сигнального плану для задачі управління світлофором зазвичай формулюється у форматі оптимізаційної задачі.

Проблема оптимізації складається з трьох компонентів: цільової функції, обмежень і змінних рішень.

Цільова функція зазвичай мінімізує або максимізує показник продуктивності системи. Операційна мета системи керування дорожнім рухом полягає в тому, щоб покращити ефективність руху на дорожній ділянці. Для оцінки використовувалися різні показники ефективності налаштування сигналів світлофора системи керування дорожнім рухом, наприклад затримка та довжина черги.

Затримка – це кількість транспортних годин, протягом яких транспортні засоби чекали на червоне світло.

Довжина черги є ще одним показником продуктивності, коли враховується кількість транспортних засобів, які не проїдуть до кінця зеленого сигналу.

Інші показники ефективності, такі як використання фази та зелена смуга пропускання, також відображають ефективність налаштувань сигналу на перехрестях.

Обмеження задачі оптимізації визначають діапазон можливих рішень. Обмеженнями для проблеми керування сигналом можуть бути динаміка руху та налаштування сигналів світлофора. Обмеження динаміки руху описують, як рух може поширюватися в дорожній мережі. Наприклад, на дорозі може бути обмеження швидкості 60 км/год, а пропускна здатність дороги може становити 230 автомобілів на 2 кілометри.

Обидві властивості дороги можна визначити в задачі оптимізації за допомогою обмежень. З точки зору налаштувань світлофора, сигнальний контролер може мати час циклу, який дорівнює 100 секундам. У цьому випадку план сигналу світлофора повинен повторюватися кожні 100 секунд. Це також необхідно визначити в задачі оптимізації за допомогою обмежень. Змінні рішення формують розв'язання задачі оптимізації.

Затримка є одним із найпопулярніших показників ефективності, який використовується як цільова функція для задачі керування сигналами світлофора. Він описує втрату ефективності в системі керування трафіком у випадках як недостатнього, так і перенасиченого. Затримка розраховується як різниця між фактичною годиною руху транспортного засобу та відстанню, пройденою транспортним засобом при швидкості вільного потоку.

Плани сигналів світлофора є змінними рішення для проблем управління сигналами світлофора. Під час роботи сигналів світлофора одиницею часу зазвичай є секунда, що означає, що всі змінні рішення є додатними цілими числами. У цьому випадку додаткові цілочисельні обмеження додаються до задачі управління світлофором, дана задача відома, як змішана цілочисельна оптимізаційна задача [9, 13, 27, 36, 37, 39].

Точні методи – це класичні методи оптимізації, де гарантовано знаходяться оптимальні рішення. Точні методи досліджують увесь спектр можливих рішень, і цей процес також відомий як пошук грубою силою або вичерпний пошук.

Прикладами двох точних методів є метод розгалуження та межі та метод січної площини. Розгалуження та межі шукають глобальний оптимальний розв’язок задачі оптимізації. Метод базується на обчисленні верхньої та нижньої межі оптимального рішення, доки значення двох меж не стануть достатньо близькими. Використовуючи верхню та нижню межі, можна зменшити кількість досліджуваних рішень.

Методу січної площини, призначений для цілочисельного програмування. Коли змінні рішення задачі оптимізації є повністю або частково цілими числами, оптимізація є проблемою цілочисельного програмування. Метод починає розв’язувати задачу цілочисельного програмування, розглядаючи змінні рішення як безперервні змінні. Створюється розріз (обмеження), щоб відокремити нецілочисельне оптимальне від усіх можливих цілочисельних рішень. Цей процес повторюватиметься до тих пір, поки не буде знайдено оптимальне ціле число.

Переваги точних методів полягають у їх простоті реалізації та точності розв’язування. Однак простота та точність призводять до того, що процес пошуку займає багато часу та ресурсів. Точний метод підходить, коли задача оптимізації має меншу кількість можливих рішень. Він не популярний у реальних проблемах, які є складними та мають великі розміри. Це також є причиною, чому він не був широко реалізований для роботи зі світлофорами.

Компанія Gartner сформулювала проблему управління сигналом (зміщенням) світлофора в моделі лінійного програмування зі змішаним цілим числом. Цільовою функцією є максимізація смуги пропускання (зелена хвиля) між скоординованими сигналами світлофора. Використовуються як точний метод, так і евристичний метод, де евристика працює приблизно в 100-300 разів швидше за час виконання без значного погіршення якості її рішення [12, 21, 27].

На відміну від точних методів, евристичні методи використовуються для вирішення складної задачі за обмеженого часу. Евристичні методи шукають не глобальні оптимальні, а близькі до оптимальних рішень. Евристичні методи

жертвують якістю рішення заради економії робочого часу, і вони мають менше обмежень під час моделювання задачі оптимізації. Деякі приклади популярних евристичних методів, які використовуються для вирішення проблем керування трафіком, — це генетичний алгоритм, оптимізація мурашиної колонії, пошук табу та моделювання відпалу.

2.3 Евристичні алгоритми керування транспортними потоками

Враховуючи вище наведені методи оптимізації, розглянемо евристичні алгоритми керування транспортними потоками, а саме: генетичний алгоритм, оптимізація мурашиної колонії, пошук табу та моделювання відпалу.

Генетичний алгоритм — це евристичний пошук, який імітує процес природного відбору. Він використовує існуючі «хороші» рішення для повторного створення нових рішень, поки не буде досягнуто майже оптимальне рішення.

Генетичний алгоритм — це популяційний підхід, коли група рішень оцінюється разом для кожного моменту регенерації. Процес регенерації включає: розмноження, кросингвер і мутацію. Вони гарантують, що приклади кращих рішень можуть бути передані наступному поколінню рішень. Процес регенерації виконуватиметься ітеративно, доки не буде виконано критерій припинення (наприклад, максимальна кількість ітерацій або лише обмежене покращення рішень).

Блок-схема генетичного алгоритму зображена на рисунку показана на малюнку 2.5. Дослідники Ло та Чоу застосували генетичний алгоритм для отримання планів синхронізації сигналу, які мінімізували загальну затримку міської артеріальної мережі.

BALANCE — це існуюча система керування трафіком, яка використовує генетичний алгоритм для оптимізації керування сигналу. Дана система працює в Гамбурзі, Інгольштадті та інших містах Німеччини [14, 25, 27, 45].



Рисунок 2.5 – Блок-схема генетичного алгоритму

Розглянувши більш детально генетичний алгоритм, перейдемо до наступного евристичного алгоритму оптимізації, який застосовується для керування транспортними потоками, а саме до алгоритму оптимізації мурашиної колонії.

Оптимізація мурашиної колонії базується на поведінці мурах, коли вони шукають шлях між своєю колонією та їжею. На початку процесу пошуку мурахи можуть вибирати випадкові шляхи для пошуку їжі. Вони відкладають феромон на своєму шляху до їжі, і шлях із вищою концентрацією феромону привабить більше мурах. У цьому випадку шлях із меншою відстанню матиме більше феромонів, оскільки для його проходження потрібно менше часу. Феромон з часом випаровується, щоб допомогти прокласти шлях із найкоротшою відстанню [19, 27].

Оптимізація колонії мурах — це евристичний пошук на основі популяції, де кожен шлях мурахи є рішенням.

Оптимізацію мурашиної колонії порівнювали з генетичним алгоритмом для вирішення проблеми керування сигналами світлофора в умовах перенасиченого руху [24, 27].

Результати показують, що оптимізація мурашиної колонії може перевершити генетичний алгоритм, якщо часу на обчислення достатньо. Іншими словами, генетичний алгоритм може бути більш придатним, коли час обчислень обмежений.

Метод мурашиних колоній також використовувався для контролю автономних перехресть і вирішення проблеми динамічного маршрутизації транспорту [18, 26, 27].

Блок-схема оптимізації мурашиної колонії зображена на рисунку 2.6.

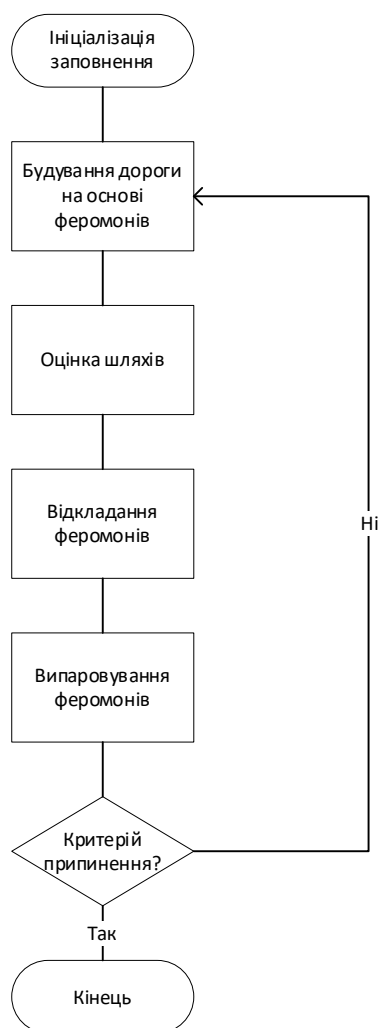


Рисунок 2.6 – Блок-схема оптимізації мурашиної колонії

Табу-пошук — це евристичний пошук єдиного рішення на основі пам'яті. Процес пошуку табу полягає у дослідженні сусідніх існуючих рішень. Список табу-пошуку — це короткочасна пам'ять, у ній зберігаються раніше відвідані рішення. Вирішуючи наступне рішення для відвідування, рішення зі списку пошуку табу будуть уникати. Це змушує пошук-табу досліджувати більше можливих просторів і запобігає процесу пошуку в пастці локального оптимуму. Найкраще рішення, запропоноване табу, буде збережено як критерій прагнення [27, 30, 40].

Критерій прагнення дозволяють табу переглянути те саме рішення, навіть якщо воно є в списку табу. Блок-схема пошуку табу зображена на рисунку 2.7. Пошук за табу використовувався для отримання планів сигналів світлофора [43].

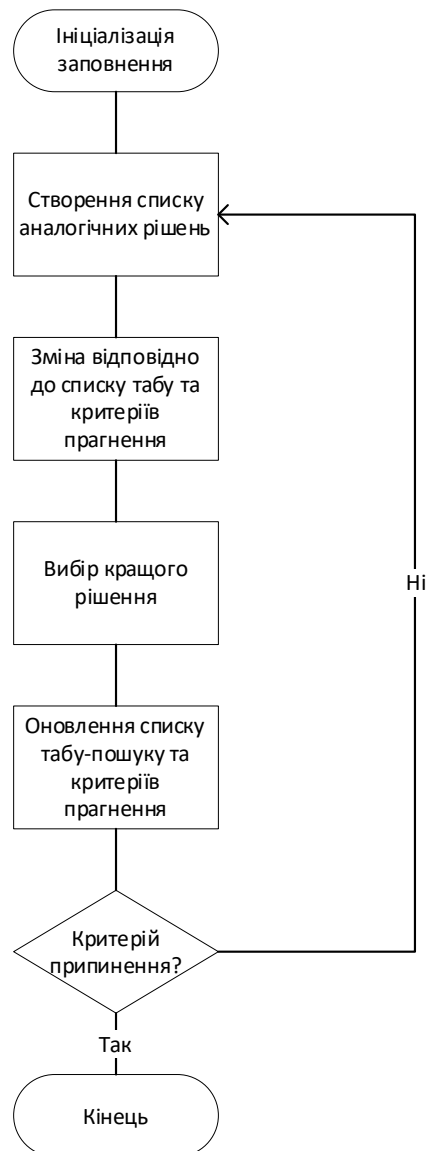


Рисунок 2.7 – Блок-схема пошуку табу

Імітований відпал — це евристичний метод пошуку, який ґрунтується на методі нагрівання та охолодження для збільшення розміру кристалів матеріалу [27, 29].

Більший розмір кристала відповідає нижчому енергетичному рівню. Спочатку матеріал нагрівається до точки плавлення, після чого починається процес охолодження. Під час процесу охолодження довший час охолодження забезпечує кращий результат.

З точки зору проблеми оптимізації, різні стани відносяться до можливих рішень, рівень енергії - це вартість можливих рішень. При певній температурі алгоритм моделювання відпалу досліджує різні стани, щоб знайти варіант із найнижчими витратами.

Якщо новий стан має нижчий рівень енергії, новий стан буде прийнято як поточний. Якщо новий стан має вищий рівень енергії, ніж поточний стан, його можна прийняти, але це залежить від ймовірності. Змодельований відпал приймає гірше рішення, яке дозволяє уникнути пасток пошуку на локальному оптимумі. Після досягнення мінімальної глобальної енергії для поточної температури, температура буде знижена, і процес пошуку повториться.

Алгоритм змодельованого відпалу нещодавно використовувався для вирішення проблеми керування сигналом світлофора, яка враховує наступні параметри керування: послідовність етапів, тривалість етапу та крок [27, 35].

Блок-схема моделювання відпалу наведена на рисунку 2.8.

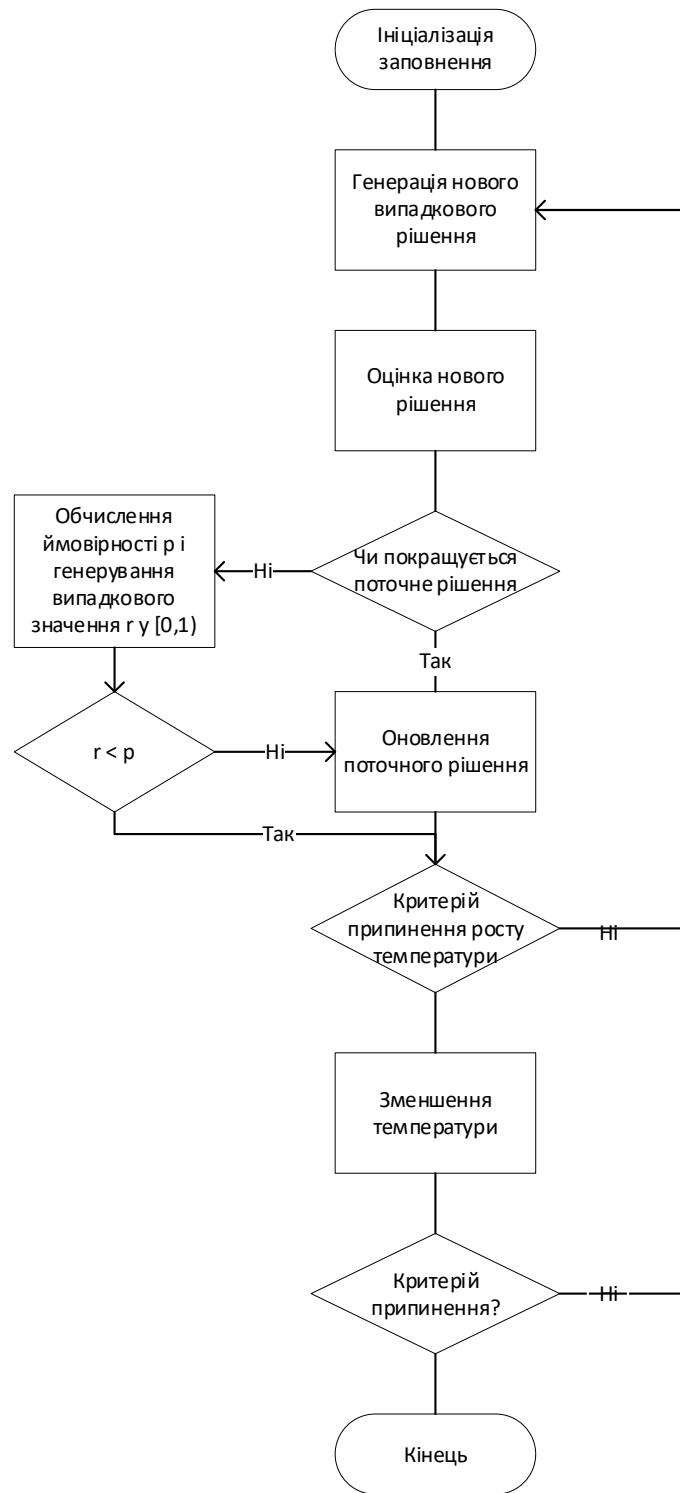


Рисунок 2.8 – Блок-схема алгоритму імітації відпалу

Враховуючи вище наведену інформацію, можемо виділити різні методи та алгоритми оптимізації, які використовуються в централізованих системах керування трафіком для отримання оптимальних даних для курування сигналом. Точний метод і евристичний метод є двома основними типами алгоритмів.

Точний метод шукає глобальне оптимальне рішення та досліджує всі можливі рішення.

Однак, як правило, це вимагає тривалого обчислювального часу, і його важко реалізувати в реальних життєвих задачах. Евристичний метод шукає локальні оптимальні рішення. Він потребує менше часу на виконання та може бути застосований до більш широкого кола проблем оптимізації. Евристичний метод більш практичний у використанні, коли час обмежений і глобальне оптимальне рішення не потрібне. Такі методи, як пошук табу, потребують пам'яті комп'ютера для запису списку рішень, які вони використали. У процесі пошуку список буде збільшуватися, оскільки список відстежує рішення з низькою продуктивністю.

Метод моделювання відпалу використовує пошук по одній точці. Через розмір проблеми керування світлофором імітований відпал може бути надто повільним для дослідження великого простору рішення.

2.4 Система оптимізації транспортних потоків

Проаналізувавши наявні розробки систем контролю методи та алгоритми оптимізації дорожнього руху, можемо виділити те, що до проблеми керування дорожнім рухом потрібно підходити комплексно, та враховувати усі технічні можливості та їх недоліки.

Зважаючи на сучасний розвиток дорожньої інфраструктури, а також проаналізувавши основні аспекти управління дорожнім рухом, технічні можливості, було розроблено схему системи контролю дорожнім рухом, структурну схему якої наведено на рисунку 2.9.

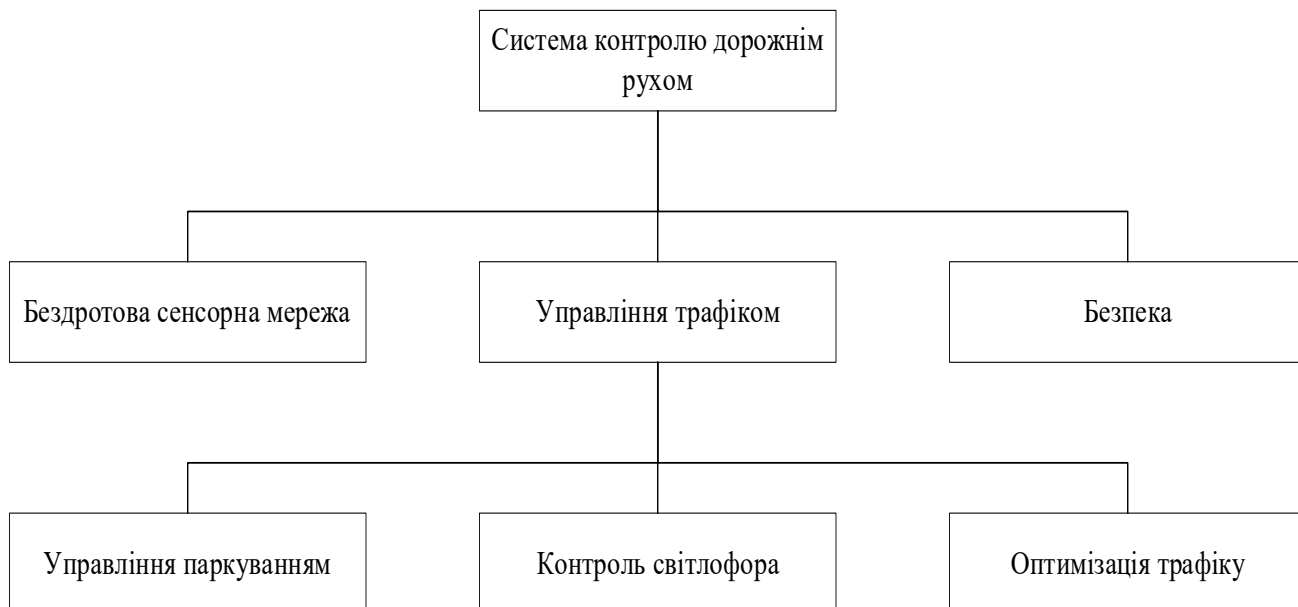


Рисунок 2.9 – Структура схеми контролю дорожнім рухом

Дана структура схеми контролю дорожнім рухом детально описує усі основні аспекти контролю дорожнім рухом. Як можна побачити з вище наведених блоків, для оптимізації транспортних потоків на автострадах важлива в першу чергу безпека на автостраді, що досягається завдяки управлінню трафіком.

Задача управління паркуванням є не менш важливою, але для досягнення оптимізації транспортних потоків на автострадах, даний блок враховуватись не буде.

Основну увагу для розробки системи оптимізації транспортних потоків, буде приділено саме контролю світлофора та оптимізації трафіку, адже дані блоки є взаємозв'язані між собою і оптимізація трафіку на пряму залежить від контролю світлофора. Існують інші фактори, які впливають на трафік транспортних потоків, такі як кількість транспорту в різний час, погодні умови, дорожньо-транспортні пригоди, але для створення системи оптимізації транспортних потоків дані фактори враховані не будуть.

Враховуючи вище наведену схему контролю дорожнім рухом, було розроблено структуру системи управління транспортними потоками схему якої наведено на рисунку 2.10.

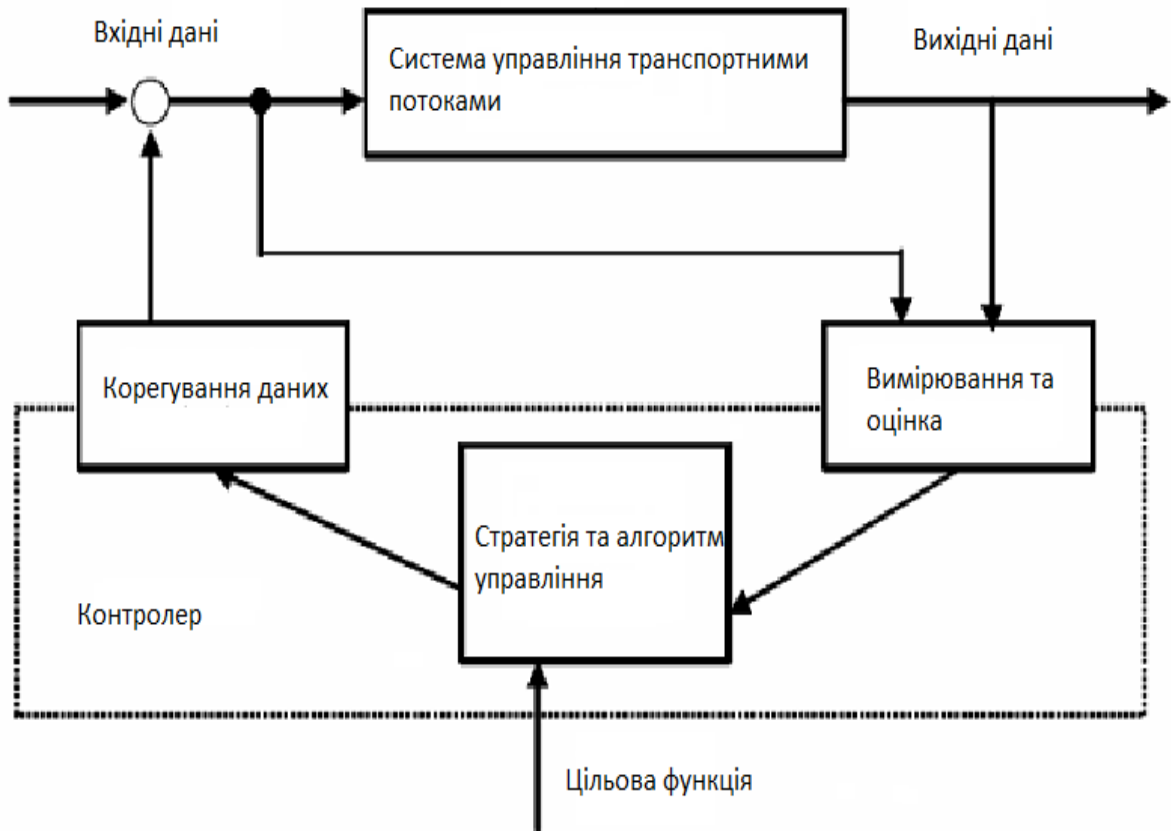


Рисунок 2.10 – Структура схеми системи управління транспортними потоками

Розглянемо більш детально схему системи управління транспортними потоками. Для вирішення задачі оптимізації управління транспортними потоками ми визначаємо цільову функцію, а саме управління світлофором за допомогою контролера на основні вхідних даних.

Спираючись тільки на алгоритм та стратегію управління контролером ми можемо виділити один вагомий недолік, дорожній рух на одній і тій самій ділянці дороги буде відрізнятися в різні пори року, в різні погодні мови, в різний час дня, крім цього не можна викреслювати дорожньо-транспортні пригоди. Саме для вирішення даної проблеми пропонується перед встановлення системи керування провести аналіз транспортних потоків за допомогою комп'ютерного зору та порахувати кількість транспортних засобів у потрібний для нас час.

2.5 Алгоритм оптимізації транспортних потоків на автострадах

Для автоматизованого управління світлофорами на дорожній ділянці будуть використані дані ультразвукового датчика відстані. Структуру системи автоматизованого управління світлофором на основі руху автотранспорту можна описати у три етапи етапи зображені на рисунку 2.11.



Рисунок 2.11 – Основні етапи роботи автоматизованого управління світлофором

Зважаючи на вищеописані етапи, а саме отримання та обробки сигналу та зміна сигналу світлофору, розробимо блок схему розглянемо блок схему системи автоматизованого управління світлофором на основі руху автотранспорту рисунок 2.12.

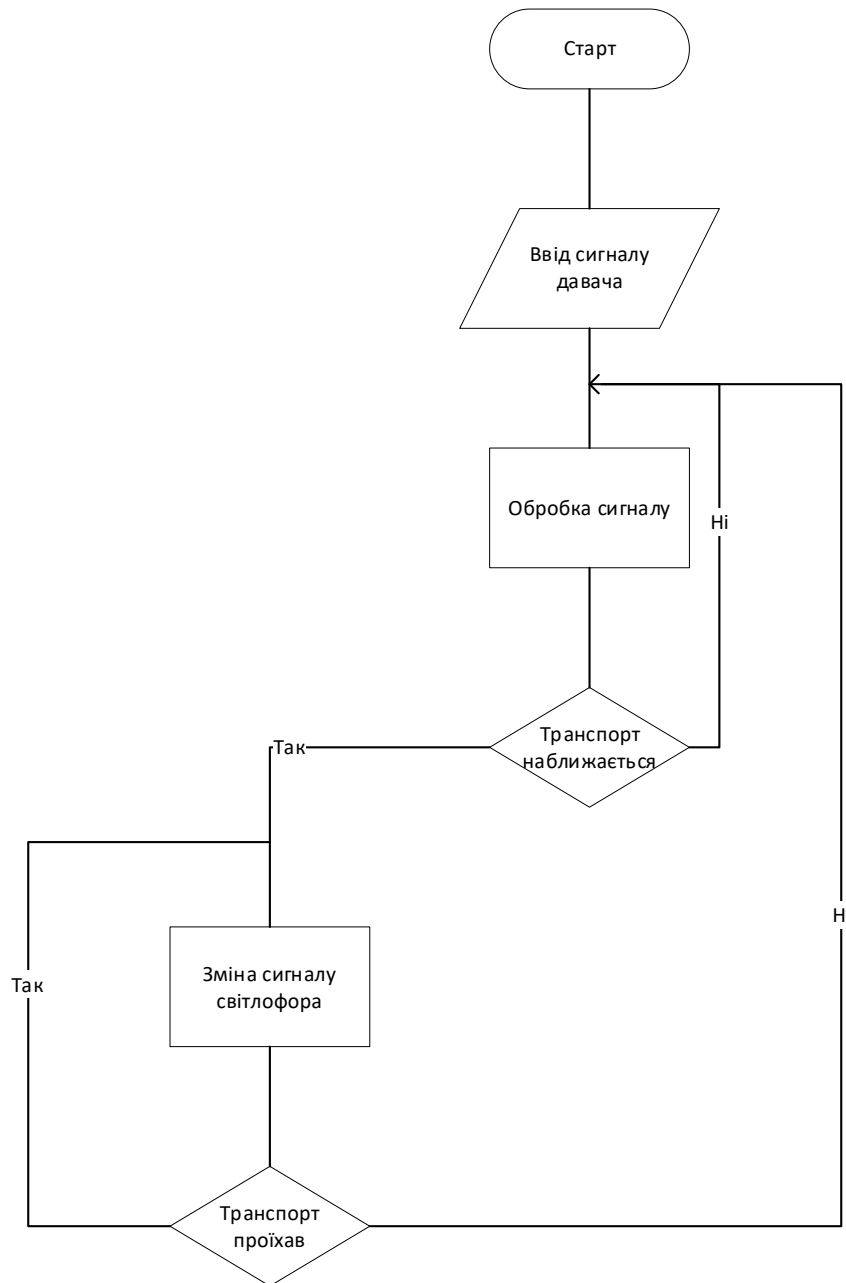


Рисунок 2.12 – Блок схема автоматизованого управління світлофором

Основна мета будь-якого класифікатора досягається на основі наданих йому вхідних даних. Вхідні дані для системи автоматизованого управління світлофором на основі аналізу руху автотранспорту – сигналів отримані давачам відстані.

Вище наведений алгоритм є хорошим для одностороннього руху без перехресть. Загалом алгоритм можна поділити на три основні кроки. Якщо на всіх напрямках перехрестя є рух, система працюватиме в штатному режимі, контролюючи сигнали по черзі.

Якщо на одній з доріг перехрестя немає руху, система пропустить цей сигнал і перейде до наступного. Наприклад, якщо на напрямках 2 та 3 немає транспортного засобу, система дозволяє проїхати транспортним засобам на напрямку 1. Потім після напрямку 1 система перейде до напрямку 4, пропускаючи напрямки 2 і 3.

Якщо на всіх 4 напрямках немає руху, система зупиниться на поточному напрямку та переміститься лише на наступний, якщо на будь-якому напрямку буде рух [27, 46]. Оптимізований алгоритм управління світлофором на перехресті зображений на рисунку 2.13.

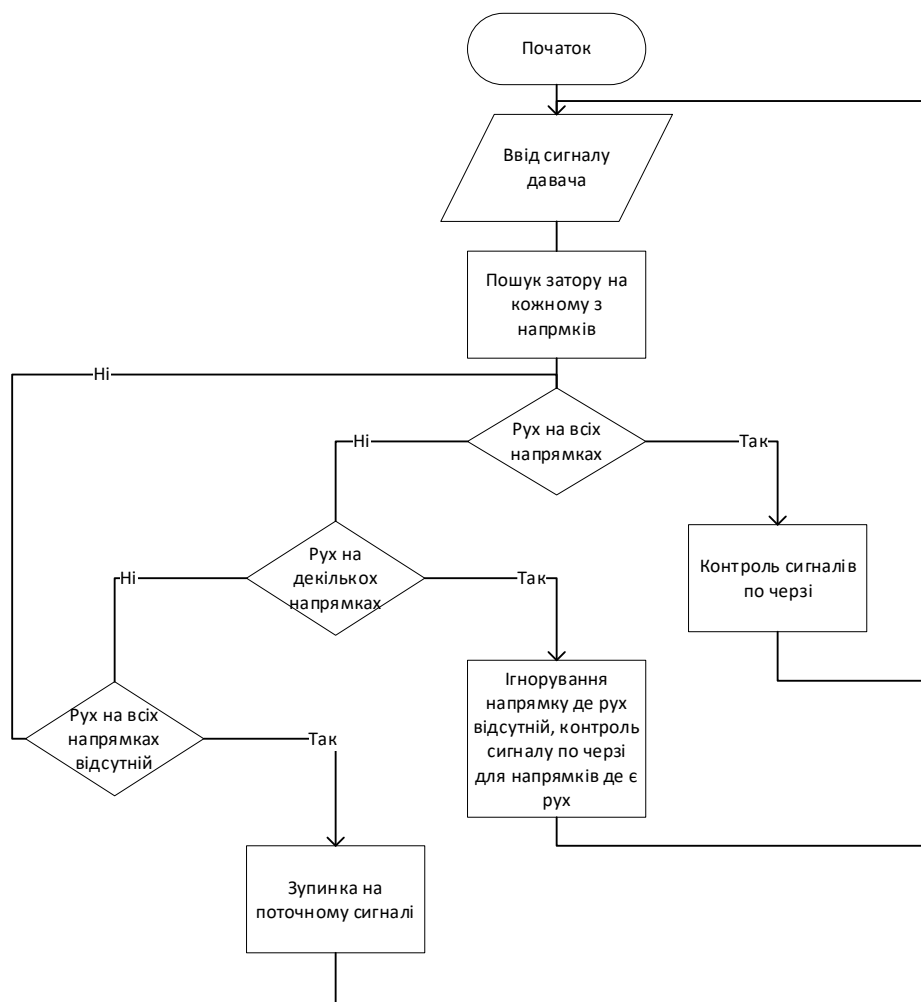


Рисунок 2.13 – Блок схема оптимізованого алгоритму автоматизованого управління світлофором

Розробивши алгоритм автоматизованого управління світлофором, потрібно зрозуміти, де доцільно даний алгоритм застосовувати, для цього використаємо технологію комп'ютерного зору, щоб виявити та підрахувати кількість транспортних засобів на потрібній ділянці у різний проміжок часу [1, 48].

Алгоритм підрахунку транспорту зображено на рисунку 2.14.

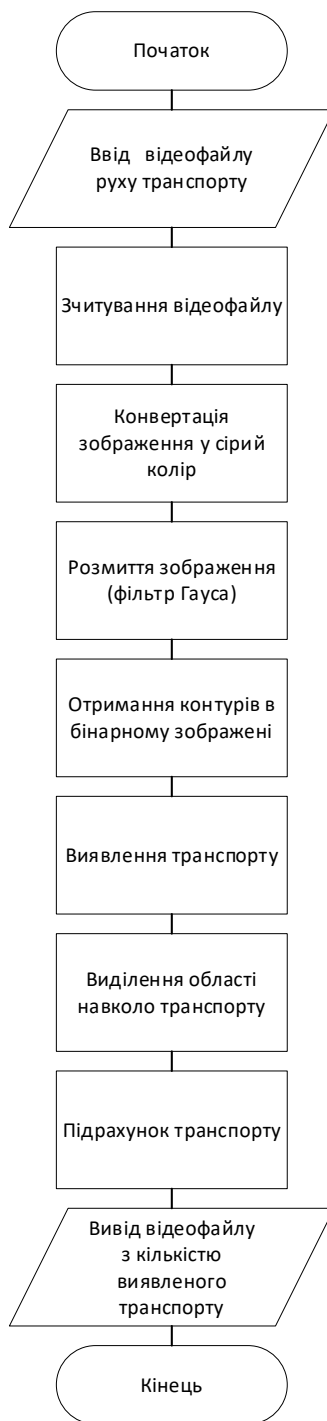


Рисунок 2.14 – Блок схема алгоритму підрахунку транспорту

За допомогою алгоритму підрахунку транспорту, буде підрахована загальна кількість автотранспорту на проблемній ділянці, у разі виникнення регулярних заторів, буде використаний засіб управління світлофорами, що і зробить можливим оптимізацію транспортних потоків на автострадах.

2.6 Висновки до розділу

Розроблено структурну схему оптимізації транспортних потоків на основі контролера та давача, що визначає наближення автотранспорту. Проаналізовано системи керування дорожнім рухом, визначено їх переваги та недоліки. Описано методи оптимізації регулювання дорожнім рухом. Проведено аналіз евристичних алгоритмів керування транспортними потоками та на основі аналізу методів та алгоритмів керування транспортними потоками, створено систему оптимізації транспортних потоків. Розроблено два алгоритми, задачею, яких є ідентифікації автотранспортних засобів, їх підрахунку для виявлення проблеми на обраній ділянці автостради та автоматизованого керування сигналами світлофорів.

3 РЕАЛІЗАЦІЯ АЛГОРИТМУ ОПТИМІЗАЦІЇ ТРАНСПОРТНИХ ПОТОКІВ НА АВТОСТРАДАХ

3.1 Структура програмного модуля системи виявлення та підрахунку транспорту на автострадах

Для реалізації програмного модуля системи виявлення та підрахунку транспорту на автострадах, який ґрунтується на технології комп'ютерного зору розроблено програмний засіб, діаграму використання, якого зображено на рисунку 3.1.

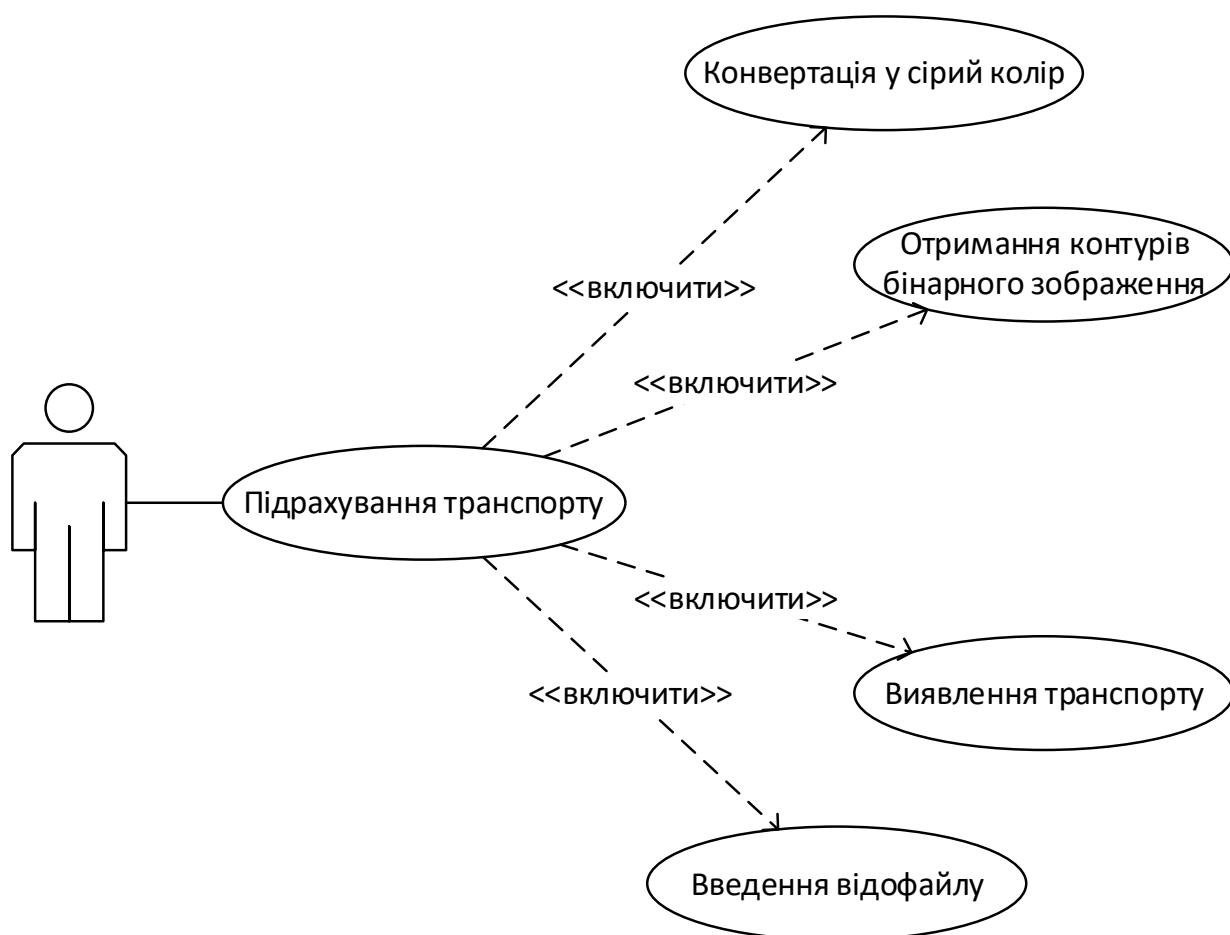


Рисунок 3.1 – Діаграма варіантів використання програмного засобу

Адміністратором розробленої програмної системи, зможуть бути працівники поліції, державного агентства автомобільних доріг України, а також працівники комунальних підприємств, які відповідають за рух транспорту у

громадах. Адміністратор матиме можливість вводити вхідні дані (відеофайл), або ж використовувати програмне забезпечення в реальному часі, на основі даних програмного засобу, буде проведено аналіз відеофайлу з ділянки дороги для виявлення та підрахунку транспорту на автострадах.

Як показано на рисунку 3.1, програмна система надає персоналу можливість для введення відеофайлу, що необхідно для реалізації алгоритму виявлення та підрахунку транспорту на автострадах за допомогою технології комп'ютерного зору.

Функція “Підрахування транспорту” містить підфункції “Конвертація у сірий колір”, “Отримання контурів бінарного зображення”, “Виявлення транспорту” та “Введення відеофайлу”.

Підфункція “Конвертація у сірий колір”, процес конвертації кольорового (або в градаціях сірого) відеофайлу в двоколірне чорно-біле, задля більш точного результату та простішого процесу обробки відеофайлу.

Підфункція “Отримання контурів бінарного зображення” передбачає можливість виявлення контурів транспорту на бінарному зображенні, що дозволить програмному засобу ідентифікувати транспорт швидше та точніше.

Підфункція “Виявлення транспорту” реалізує можливість більш точно оглянути персоналу зону виявлення транспорту на дорожній ділянці за допомогою побудованого навколо транспорту прямокутного виділення, що дозволить виключити похибку в підрахунку.

Програмну систему для виявлення та підрахунку транспорту на автострадах на основі комп'ютерного зору, була розроблена на основі структурно-функціонального підходу.

Алгоритм функціонування програмного модуля, який відповідає за побудову рамки навколо виявленого автотранспорту та підрахунок наведено на рисунку 3.2.



Рисунок 3.2 – Алгоритм функціонування програмного модуля виявлення та підрахунку транспорту на автострадах

Як бачимо на рисунку 3.2, алгоритм програмного модуля виявлення та підрахунку транспорту на автострадах на основі комп'ютерного зору включає реалізацію чотирьох функцій. Функція підрахунку транспорту передбачає окрім підрахунку виведення результатів обробки відеофайлу та візуалізації знайденого автотранспорту.

Зважаючи на вищезазначене, у наступному підрозділі охарактеризуємо програмне середовище для реалізації алгоритму виявлення та підрахунку транспорту на автострадах.

3.2 Реалізація алгоритму виявлення та підрахунку транспорту на автострадах

Для реалізації алгоритму виявлення та підрахунку транспорту на автострадах використаємо мову програмування Python.

Python — це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Його високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання як мови сценаріїв або з'єднувальної мови для існуючих компонентів. Простий, легкий для вивчення синтаксис Python підкреслює читабельність, а отже, знижує вартість обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду.

Інтерпретатор Python і велика стандартна бібліотека доступні у вихідному або двійковому вигляді безкоштовно для всіх основних платформ і можуть вільно поширюватися. Бібліотеки, такі як NumPy і Matplotlib дозволяють ефективно використовувати Python в наукових обчисленнях. OpenCV має прив'язки Python з багатим набором функцій для комп'ютерного зору та обробки зображень.

Для роботи з мовою програмування Python, було обрано інтегроване середовище розробки PyCharm. PyCharm — це інтегроване середовище розробки із повним набором інструментів для розробки на Python. Крім того, інтегроване середовище розробки надає можливості для професійної веб-розробки за допомогою Django. PyCharm дозволяє писати код швидше та легше в розумному та настроюваному редакторі з автозавершенням коду, фрагментами, згортанням коду та підтримкою розділених вікон. Основними перевагами даного середовища розробки є:

- інтелектуальна підтримка кодування;
- інтелектуальний редактор коду;

- інтелектуальна навігація по коду;
- швидкий і безпечний рефакторинг;
- налагодження, тестування та профілювання;
- інструменти бази даних;
- підтримка наукових інструментів, таких як matplotlib, numpy та scipy.

Інтерфейс інтегрованого середовища розробки PyCharm достатньо простий та зрозумілий. Екранну форму загального вікна інтерфейсу середовища PyCharm наведено на рисунку 3.3.

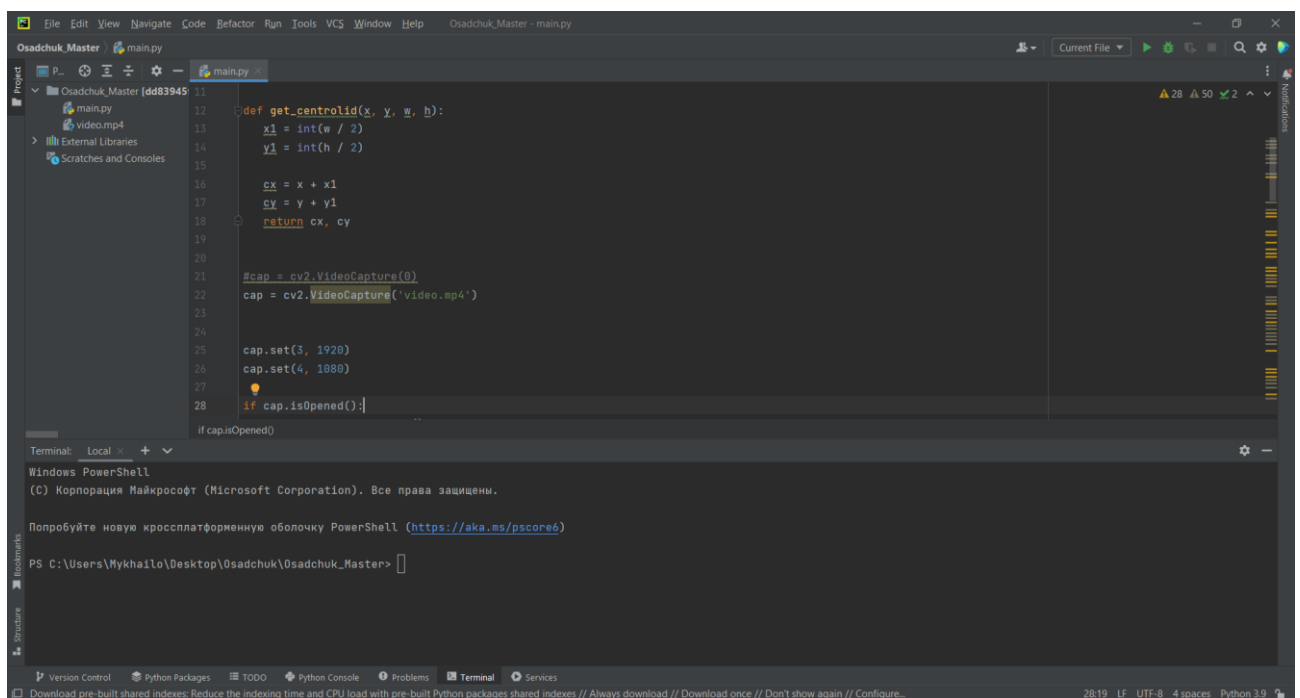


Рисунок 3.3 – Екранна форма інтерфейсу інтегрованого програмного середовища PyCharm

Для реалізації алгоритму виявлення та підрахунку транспорту на автострадах, а саме для обробки зображення було використано бібліотеки OpenCV та NumPy.

OpenCV — це бібліотека з відкритим вихідним кодом, яка використовується для розробки програм комп'ютерного бачення, таких як аналіз відео, аналіз записів із камер відеоспостереження та аналіз зображень.

Бібліотека OpenCV створена на C++ і містить понад 2500 оптимізованих алгоритмів. Під час створення програм з використанням комп'ютерного зору, є можливість не будувати програмне забезпечення з нуля, а використовувати дану бібліотеку. На даний момент більшість компаній використовують бібліотеку OpenCV, наприклад: Google, Amazon, Microsoft і Toyota. Багато дослідників і розробників роблять свій внесок. Бібліотека OpenCV легко встановлюється на будь-яку операційну систему, наприклад: Windows, Ubuntu і MacOS.

Бібліотека NumPy використовується для підтримки багатовимірних масивів, матриць тощо на мові програмування Python. Це числова бібліотека Python з відкритим кодом. Бібліотека NumPy включає:

- інструменти для інтеграції програм C/C++ і Fortran
- складні функції;
- потужний елемент N-вимірної масиви;
- лінійну алгебру, перетворення Фур'є, випадкові числа тощо.

Використовуючи вище наведені бібліотеки OpenCV та NumPy, середовище розробки PyCharm та мову програмування Python, було реалізовано програмне забезпечення для реалізації алгоритму виявлення та підрахунку транспорту на автострадах. Розглянемо більш детально використані функції для реалізації програмного засобу.

Функція VideoCapture відповідає за використання відеофайлу, або ж використання відеокамери в реальному часі.

Функція VideoCapture.isOpened() повертає істину, якщо відео вже ініціалізовано, і якщо попередній виклик конструктора VideoCapture завершився успішно, метод повертає істину.

VideoCapture.read() повертає відеокадр. Якщо жодного кадру не було захоплено, зображення буде порожнім.

Функція cv2.cvtColor перетворює зображення з одного колірної простору в інший. У разі переходу до/з колірної простору RGB порядок каналів слід вказати точно (RGB або BGR). Потрібно зауважити, що формат кольору за замовчуванням у OpenCV часто називають RGB, але насправді це BGR (перевернуті байти).

Функція `cv2.dilate()` розширює зображення за допомогою певного структурного елемента. Функція розширює вихідне зображення, використовуючи вказаний структурний елемент, який визначає форму околиці пікселя.

Функція `cv2.threshold()` застосовує порогове значення фіксованого рівня до кожного елемента масиву. Дана функція зазвичай використовується для отримання дворівневого (бінарного) зображення із зображення у градаціях сірого або для видалення шуму, тобто фільтрації пікселів із занадто малими або занадто великими значеннями. Функція підтримує кілька типів порогових значень. Вони визначаються параметрами типу.

Функція `cv2.GaussianBlur()` розмиває зображення за допомогою фільтра Гауса. Функція згортає вихідне зображення з указаним ядром Гауса. Підтримується фільтрація на місці.

Функція `cv2.getStructuringElement()` повертає структурний елемент заданого розміру та форми для морфологічних операцій. Функція створює та повертає структурний елемент, який можна далі передати.

Функція `cv2.morphologyEx()` може виконувати розширені морфологічні перетворення, використовуючи ерозію та розширення як основні операції. Будь-яка з операцій може бути виконана на місці. У випадку багатоканальних зображень кожен канал обробляється незалежно.

Функція `cv2.findContours()` знаходить контури в бінарному зображенні за допомогою алгоритму. Контури є корисним інструментом для аналізу форми, виявлення та розпізнавання об'єктів.

Функція `cv2.boundingRect(масив)` обчислює правий верхній обмежувальний прямокутник набору точок або ненульових пікселів зображення в градаціях сірого. Функція обчислює та повертає мінімальний верхній правий обмежувальний прямокутник для вказаного набору точок або ненульових пікселів зображення в градаціях сірого [1, 48].

Для запуску програмного забезпечення потрібно використати команду: «`python main.py`» у терміналі. Приклад роботи програми з конвертованим відеофайлом зображено на рисунку 3.4.



Рисунок 3.4 – Приклад роботи програми з конвертованим відеофайлом

Програмний засіб не тільки конвертує зображення, як наведено вище, а й виявляє та рахує автотранспорт, приклад роботи даних функцій зображено на рисунку 3.5.

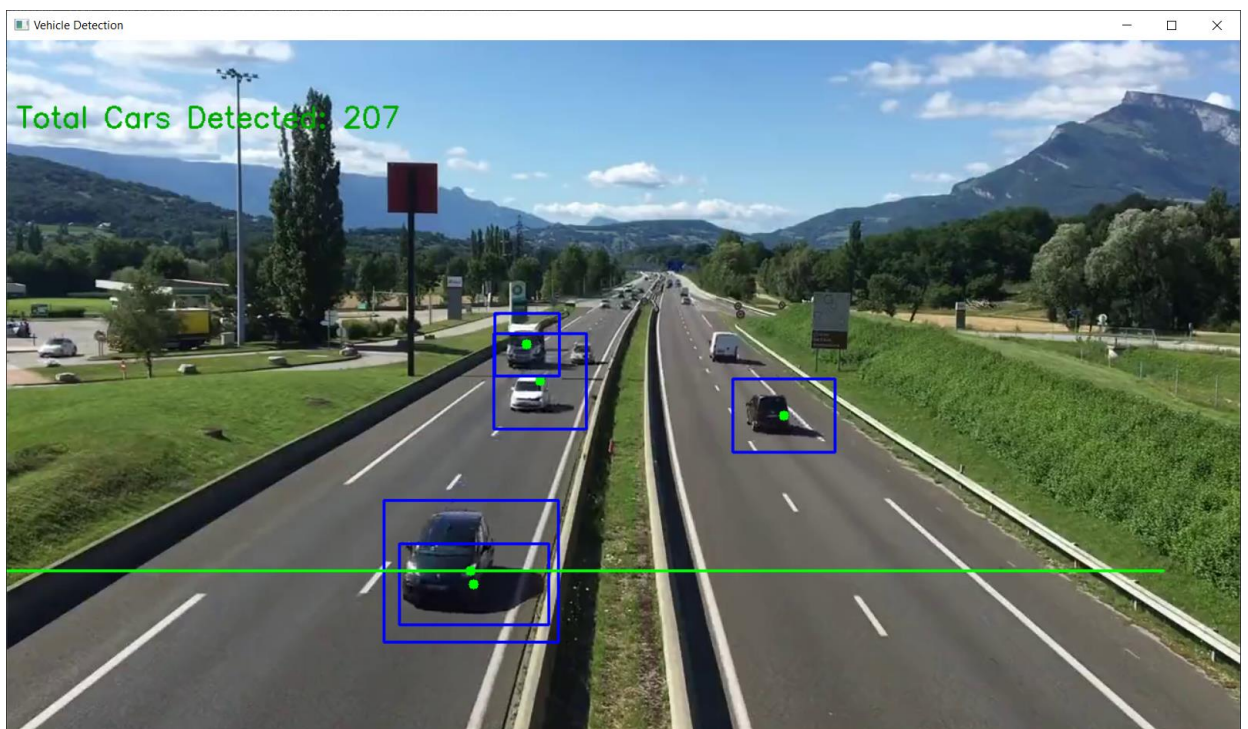


Рисунок 3.5 – Приклад роботи програми виявлення та підрахунку транспорту

Для перевірки точності роботи програмного засобу, а виявлення та підрахунку транспорту, було проведено кілька досліджень. Для дослідження було використано 10 відеофайлів, з них для дослідження було використано перші 60 секунд. Для встановлення точності дані, було використано людський ресурс, для мануального підрахунку транспорту. Результат досліджень точності роботи програмного засобу зображено на рисунку 3.6.



Рисунок 3.6 – Діаграма точності підрахунку транспорту

Зважаючи на діаграму точності постановки діагнозу за допомогою програмного засобу виявлення катаракти на основі медичних зображень, можна зрозуміти, що точність підрахунку транспорту складає 89%, що є недостатнім для автономної роботи програмного забезпечення, але є достатнім для підрахунку та аналізу руху автотранспорту, використовуючи оператора, адміністратора, що і було ціллю програмного засобу виявлення та підрахунку транспорту.

Даний програмний засіб буде використаний, як основа для аналізу дорожнього руху на проблемній ділянці дороги, щоб оптимізувати транспортні потоки за допомогою програмно-апаратного засобу автоматизованого управління світлофорами.

3.3 Модель підсистеми автоматизованого управління світлофорами на основі мереж Петрі

Для реалізації вищеприписаного алгоритму системи автоматизованого управління світлофорами розроблено структурну модель підсистеми автоматизованого управління світлофорами на основі мереж Петрі, модель якого зображено на рисунку 3.7.

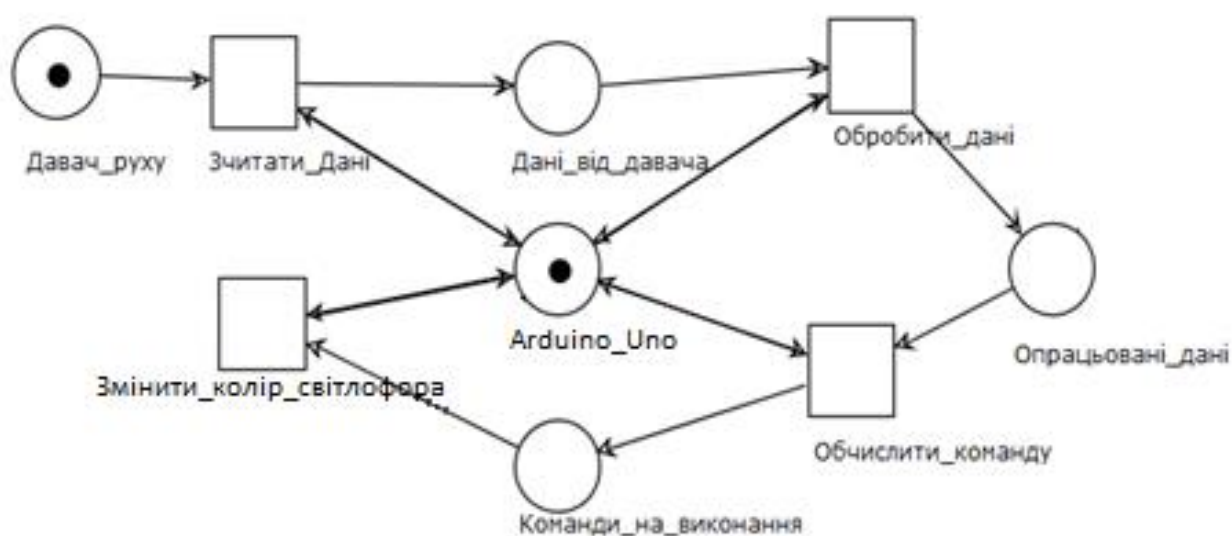


Рисунок 3.7 – Структурна модель підсистеми автоматизованого управління світлофорами на основі мереж Петрі

Враховуючи структурну модель підсистеми автоматизованого управління світлофорами на основі мереж Петрі сформуємо граф досяжності станів (при спрацюванні датчика руху), зображення графа на рисунку 3.8.

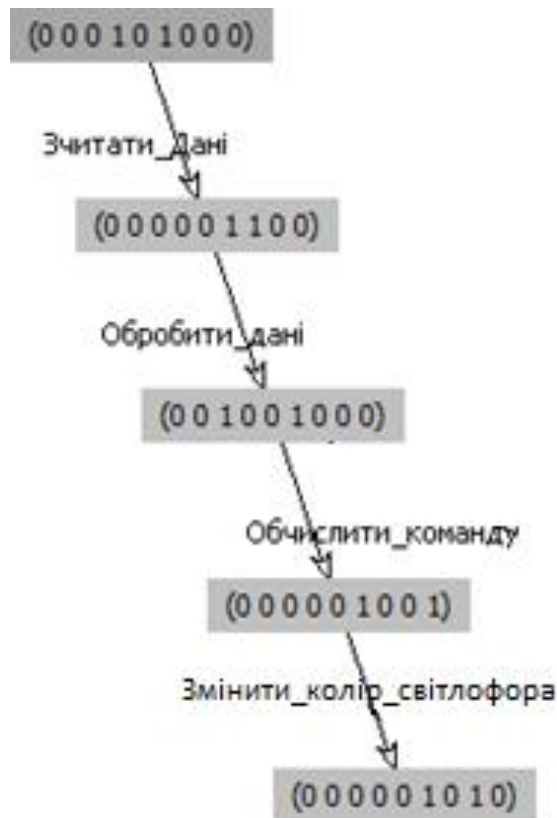


Рисунок 3.8 – Граф досяжності станів (при спрацюванні давача руху)

Зважаючи на структурну модель підсистеми автоматизованого управління світлофорами на основі мереж та граф досяжності станів (при спрацюванні давача руху), у наступному підрозділі охарактеризуємо фізичну модель системи автоматизованого управління світлофорами.

3.4 Модель системи автоматизованого управління світлофорами

Для реалізації схеми пристрою системи автоматизованого управління світлофорами на основі мікроконтролера було обрано онлайн-середовище EasyEDA, оскільки, воно має ряд плюсів, а саме:

- наявність безкоштовної версії;
- простота у використанні;
- можливість виведення друкованих плат.

EasyEDA — це веб-набір інструментів EDA, який дозволяє інженерам апаратного забезпечення проектувати, моделювати схеми та друковані плати. EasyEDA дозволяє створювати та редагувати принципові діаграми, моделювати SPICE змішаних аналогових і цифрових схем, створювати та редагувати макети друкованих плат і, за бажанням, виготовляти друковані плати.

Інші функції включають створення переліку матеріалів, файлів Gerber, а також файли вибору й розміщення та документальних виходів у форматах PDF, PNG та SVG.

На рисунку 3.9 зображено інтерфейс програми EasyEDA.

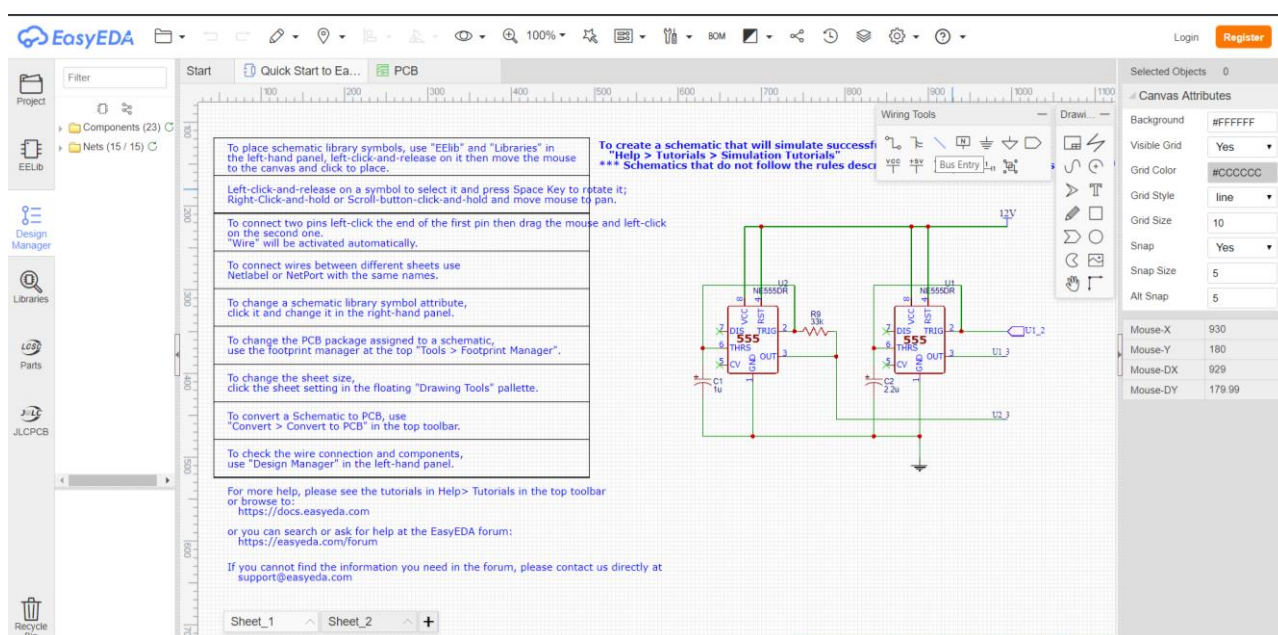


Рисунок 3.9 – Інтерфейс програми EasyEDA

Використовуючи програмний засіб EasyEDA було створено функціональну схему системи автоматизованого управління світлофорами на автостадах.

На рисунку 3.10 зображена функціональна схема системи автоматизованого управління світлофорами.

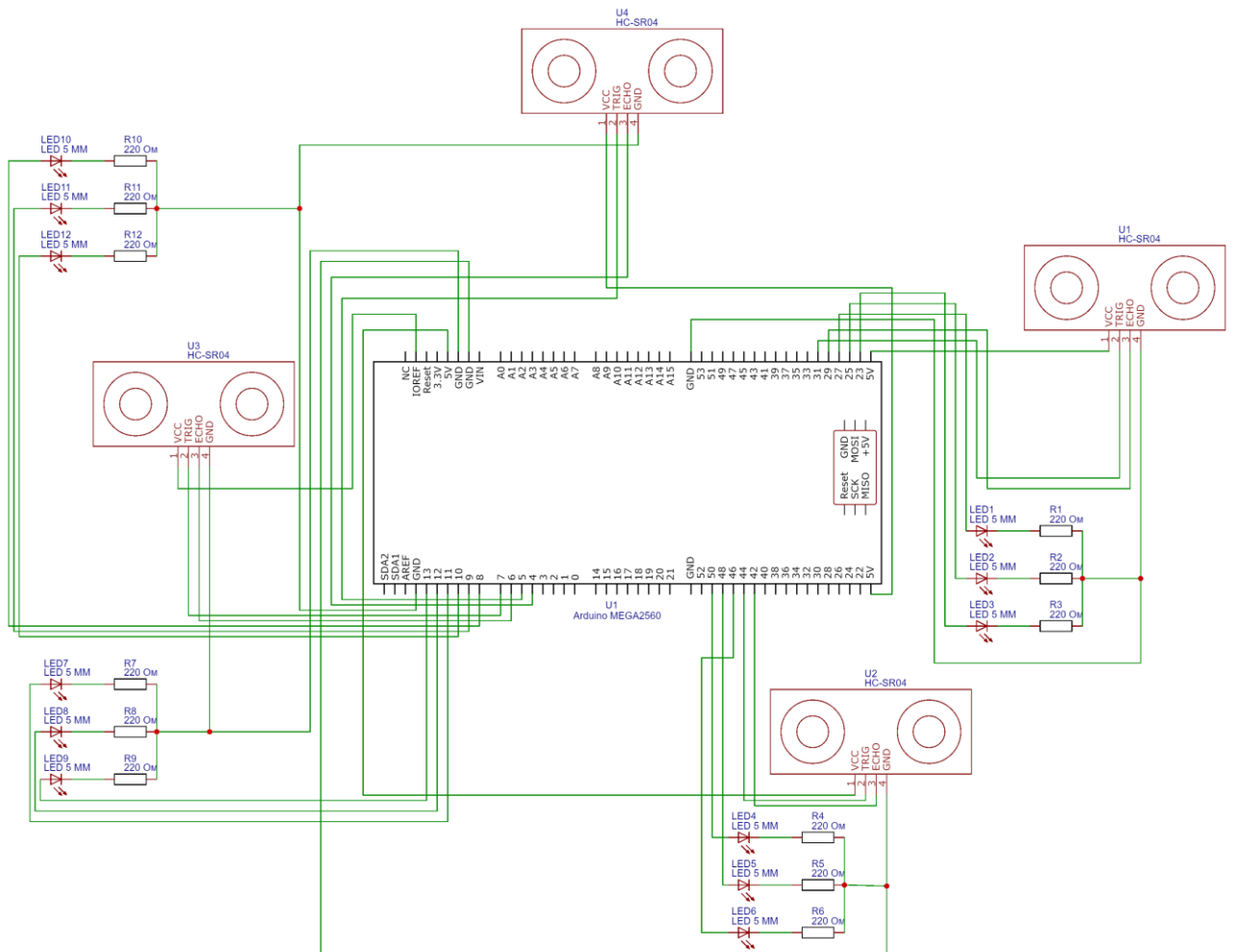


Рисунок 3.10 – Функціональна схема системи автоматизованого управління світлофорами

Для побудови макету пристрою було обрано платформу під назвою Fritzing, оскільки вона має ряд переваг:

- наявність безкоштовної версії;
- простота у використанні;
- можливість виведення друкованих плат;
- можливість автономної роботи.

Fritzing — це платформа для розробки апаратного забезпечення з відкритим кодом, яка дозволяє проектувати електронні схеми на комп'ютері. Метою програмного забезпечення є підтримка розробників у реалізації їхніх проектних ідей. Програмне забезпечення було створено членами Потсдамського університету прикладних наук.

Fritzing пропонує багато електронних компонентів у каталозі,

класифікованому відповідно до асортименту, який можна доповнити власними компонентами або компонентами інших користувачів. Ці компоненти можна вставити у віртуальну макетну плату через GUI та з'єднати один з одним у різних поданнях. Діапазон компонентів також зосереджений на друкованих платах, які належать до платформи Arduino.

Fritzing дозволяє створювати друковані плати та надає доступ до комерційної послуги, відомої як FritzingFab, щоб замовляти друковані плати, створені у програмному забезпеченні Fritzing.

На рисунку 3.11 зображено інтерфейс програми Fritzing

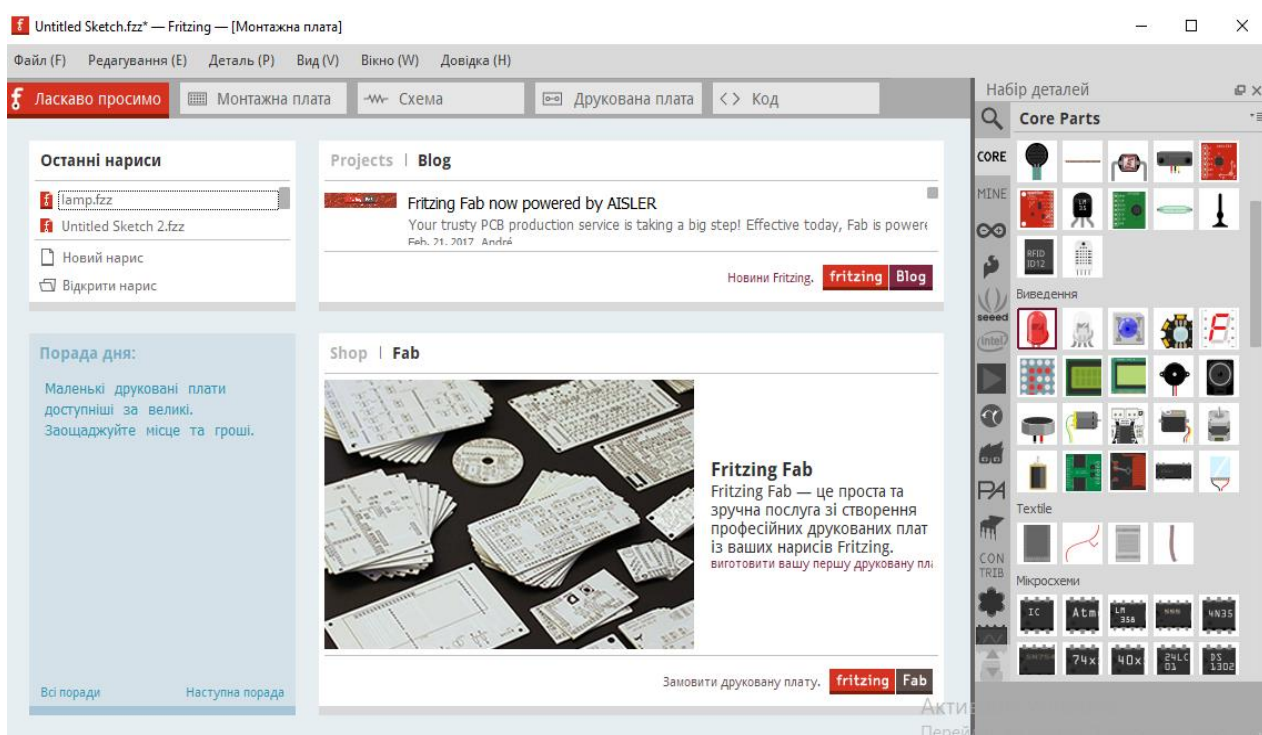
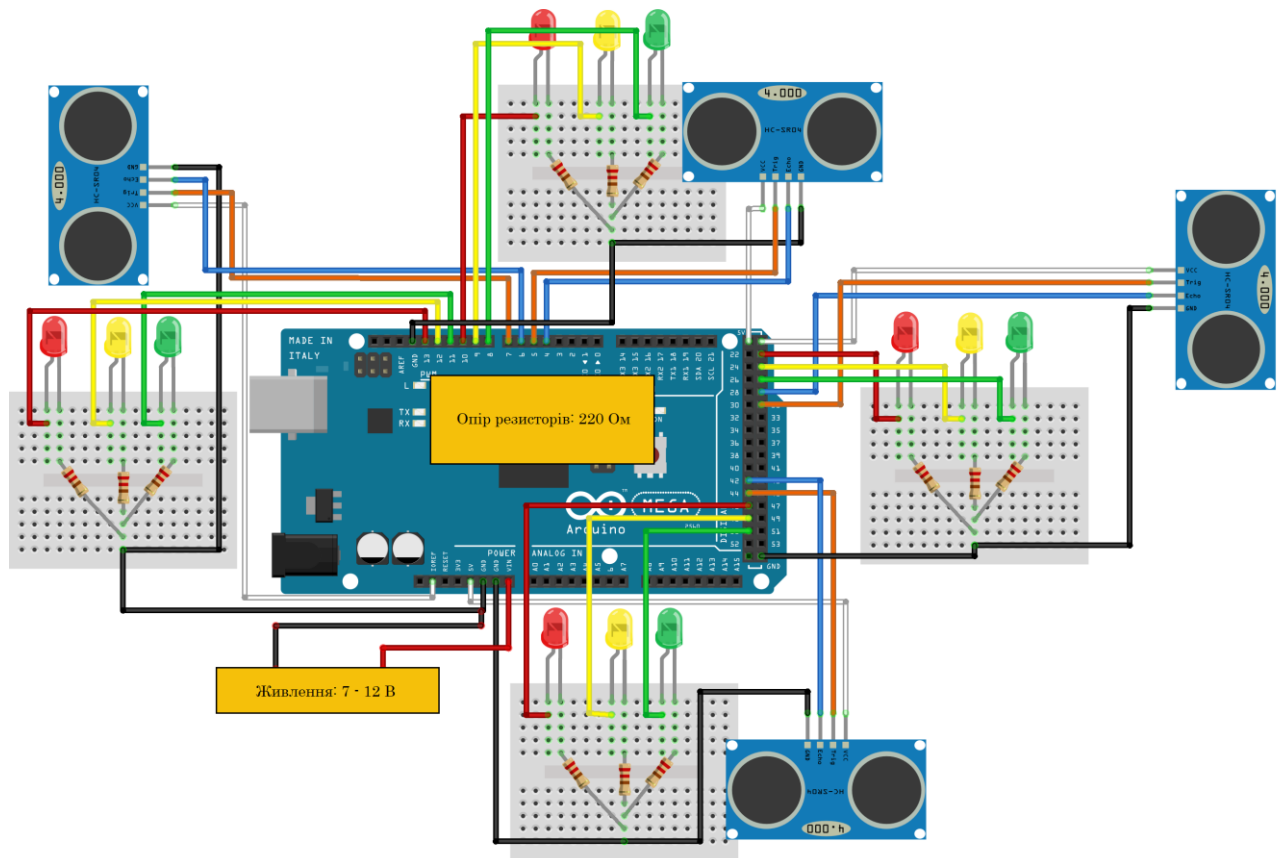


Рисунок 3.11 – Інтерфейс програми Fritzing

Використовуючи програмне середовище Fritzing було створено функціональну схему системи автоматизованого управління світлофорами на автострадах [46]. Макет системи автоматизованого управління світлофорами зображено на рисунку 3.12.



P

Рисунок 3.12 – Макет системи автоматизованого управління світлофорами

На основі макету та функціональної буде розроблена програмно-апаратна система автоматизованого управління світлофорами на основі датчика руху та аналізу транспортних потоків.

3.5 Програмно-апаратна реалізація системи автоматизованого управління світлофорами

Для програмно-апаратної реалізації системи автоматизованого управління світлофорами за основу було взято плату мікроконтролера Arduino Mega 2560 Rev3 та ультразвуковий датчик руху HC-SR04.

Arduino Mega — це плата мікроконтролера на основі ATmega2560. Mega 2560 R3 також має контакти SDA та SCL поруч із AREF. Крім того, є два нових контакти, розташовані біля контакту RESET. Одним з них є IOREF, який дозволяє

адаптуватися до напруги, що подається від плати. Інший не підключений і зарезервований для майбутніх цілей [20, 49, 50]. На рисунку 3.13 зображено мікроконтролер Arduino Mega 2560 Rev3.

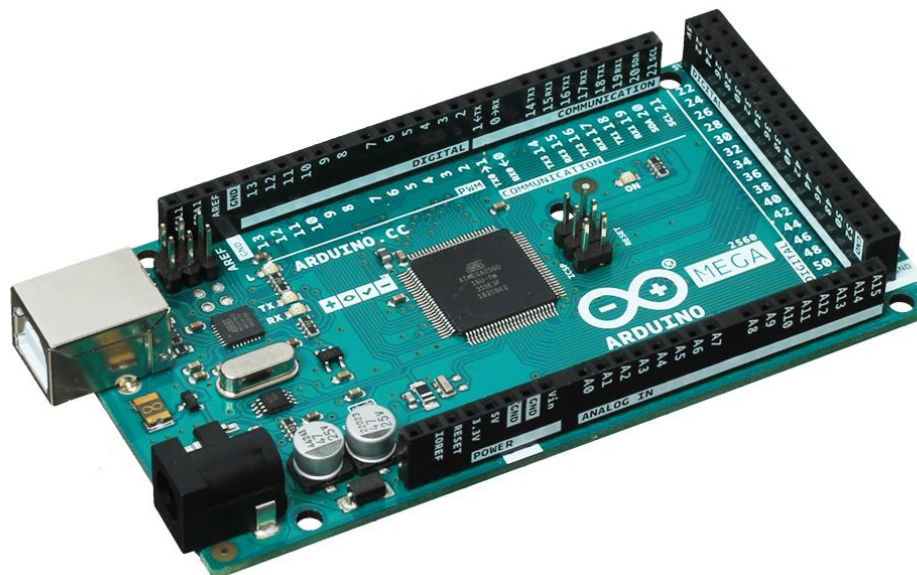


Рисунок 3.13 – Мікроконтролер Arduino Mega 2560 Rev3

Технічні характеристики мікроконтролера Arduino Mega 2560 Rev3 вказано в таблиці 3.1

Таблиця 3.1 – Технічні характеристики мікроконтролера Arduino Mega 2560 Rev3

Мікроконтролер	ATmega2560
Робоча напруга	5 В
Вхідна напруга (рекомендована)	7-12 В
Вхідна напруга (граничне)	6-20 В
Цифрові Входи / Виходи	54 (з яких 15 забезпечують вихід ШІМ)
Аналогові входи	16
Постійний струм через вхід / вихід	20 мА
Постійний струм для виведення 3.3В	50 мА
Флеш пам'ять	256 Кб з яких 8 Кб використовуються для завантажувача
ОЗУ	8 Кб
EEPROM	4 Кб
Тактова частота	16 МГц

Розглянемо застосування ультразвукових давачів руху HC-SR04, саме дані давачу будуть використані для створення системи автоматизованого управління світлофорами. Давач HC-SR04 зображено на рисунку 3.14.



Рисунок 3.14 – Модуль давача відстані HC-SR04

HC-SR04 — ультразвуковий модуль вимірювання дальності, який забезпечує функцію безконтактного вимірювання від 2 до 400 см. Точність визначення діапазону може досягати 3 мм, а ефективний кут становить $< 15^\circ$. Його можна живити від джерела живлення 5В.

Ультразвуковий давач відстані HC-SR04 складається з двох ультразвукових перетворювачів. Один діє як передавач, який перетворює електричний сигнал на ультразвукові імпульси 40 кГц. Інший діє як приймач і приймає передані імпульси. Коли приймач отримує ці імпульси, він створює вихідний імпульс, ширина якого пропорційна відстані до об'єкта попереду. Даний давач забезпечує відмінне безконтактне визначення діапазону від 2 см до 400 см з точністю 3 мм.

Модуль HC-SR04 не водонепроникний, а також може пошкодитися брудом. Але на ринку є спеціальні модулі, розроблені з певним рівнем водонепроникності та можуть використовуватися в середовищах із високою вологістю.

Оскільки давач може працювати від 5 вольт, його можна підключити безпосередньо до Arduino або будь-якого іншого логічного мікроконтролера .

Технічні характеристики давача ультразвукового давача відстані HC-SR04 вказано у таблиці 3.2

Таблиця 3.2 – Технічні характеристики ультразвукового давача відстані HC-SR04

Напруга живлення	+5 В - постійний струм
Сила струму спокою	<2 мА
Робоча сила струму	15 мА
Ефективний робочий кут	<15 °
Відстань вимірювань	від 2 см до 400 см (1-13 дюймів)
Роздільна здатність	0.3 см
Кут вимірювань	30 градусів
Ширина імпульсу тригер	10 мікросекунд
Розміри	45 мм x 20 мм x 15 мм

Для реалізації програмного забезпечення системи автоматизованого управління світлофорами було обрано середовище розробки Arduino IDE.

Arduino IDE — це програмне забезпечення з відкритим кодом, розроблене Arduino.cc і в основному використовується для написання, компіляції та завантаження коду майже до всіх модулів Arduino. Це офіційне програмне забезпечення Arduino, яке робить компіляцію коду.

Arduino IDE доступний для всіх операційних систем, наприклад MAC, Windows, Linux, і працює на платформі Java, яка має вбудовані функції та команди, які відіграють важливу роль у налагодженні, редагуванні та компіляції коду.

Основний код, також відомий як ескіз, створений на платформі IDE, зрештою створить шістнадцятковий файл, який потім передається та завантажується в контролер на платі.

Arduino IDE в основному містить дві основні частини: редактор і компілятор, де перший використовується для написання необхідного коду, а останній використовується для компіляції та завантаження коду в даний модуль Arduino. Дане середовище підтримує мови C і C++.

Екранну форму загального вікна інтерфейсу середовища Arduino IDE наведено на рисунку 3.15.

```
sketch_may25a
#define led1 11
#define led2 10

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (distance < 15) { // На цьому етапі проходить вкл/вкл світлодіода
    digitalWrite(led1,HIGH); // коли загоряється червоною, зелений об'язав вимкнутися
    digitalWrite(led2,LOW);
  } else {
    digitalWrite(led1,LOW); digitalWrite(led2,HIGH);
  }

  if (distance >= 200 || distance <= 0){
    Serial.println("Out of range"); // Вне діапазона
  }
  else {
    Serial.print(distance);
    Serial.println(" см"); // тут теж можна вказати " см"
  }
}
```

Рисунок 3.15 – Інтерфейсу середовища Arduino IDE

Враховуючи усе вище вказане, було реалізовано програмне забезпечення для системи автоматизованого управління світлофорами та вказано усі піни.

Перш за все, було включено бібліотеку TimerOne. Дана бібліотека використовується для багаторазового вимірювання періоду часу в мікросекундах, і в кінці кожного періоду буде викликати функцію переривання.

Бібліотека TimerOne була використана, тому що зчитування з датчиків і керування світлодіодами має відбуватись одночасно. Доведеться використовувати затримку між сигналами світлофора, щоб не зчитувати сигналів з датчиків відстані постійно. Тому була використана дана бібліотека, яка дозволить викликати функцію, у котрій в свою чергу будуть безперервно зчитуватись дані з датчиків, а у функції циклу стане можливо контролювати сигнали світлофора.

Функцію Timer1.initialize(). потрібно викликати перед використанням будь-якого іншого методу бібліотеки TimerOne.

«Мікросекунди» — це фактично період часу, який займає таймер. Тут необов'язково вказувати період таймера. Стандартний період становить 1 секунду. Потрібно наголосити, що це порушує analogWrite() на цифрових контактах 9 і 10.

Timer1.attachInterrupt(softInterr) викликає функцію щоразу, коли закінчується період таймера. Було встановлено період таймера на 100 000, тому дана функція буде викликана через 100 мілісекунд. Дані таймеру потрібно змінювати відповідно до результатів аналізу руху транспорту на проблемній ділянці дороги

У функції циклу перевіряється, чи є транспортні засоби на відстані 400 см і ближче чи ні. Якщо транспортний засіб виявлено, то буде викликана функція для зміни сигналу світлофора для даного транспортного засобу.

Лістинг функції циклу зображено на рисунку 3.16.

```
void loop()
{
  // Транспортні засоби є на напрямку 1
  if(S1<t)
  {
    signal1Function();
  }
  // Транспортні засоби є на напрямку 2
  if(S2<t)
  {
    signal2Function();
  }
  // Транспортні засоби є на напрямку 3
  if(S3<t)
  {
    signal3Function();
  }
  // Транспортні засоби є на напрямку 4
  if(S4<t)
  {
    signal4Function();
  }
}
```

Рисунок 3.16 – Функція циклу

Softinterr() — це функція переривання, яка викликається кожні 100 мілісекунд. У цій функції реалізовано зчитування даних ультразвукових датчиків і обчислення відстані.

Лістинг функції переривання Softinterr() зображено на рисунку 3.17.

```

void softInterr()
{
  // Зчитування з першого ультразвукового датчика
  digitalWrite(triggerpin1, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin1, LOW);
  time = pulseIn(echopin1, HIGH);
  S1= time*0.034/2;
}

```

Рисунок 3.17 – Функція переривання Softinterr()

Після написання коду, відбувається його завантаження на плату Arduino через USB кабель. Для зміни параметрів потрібно лише замінити відповідні значення в коді [46]. На рисунку 3.18 зображено зібрану автоматизовану систему управління світлофорами.

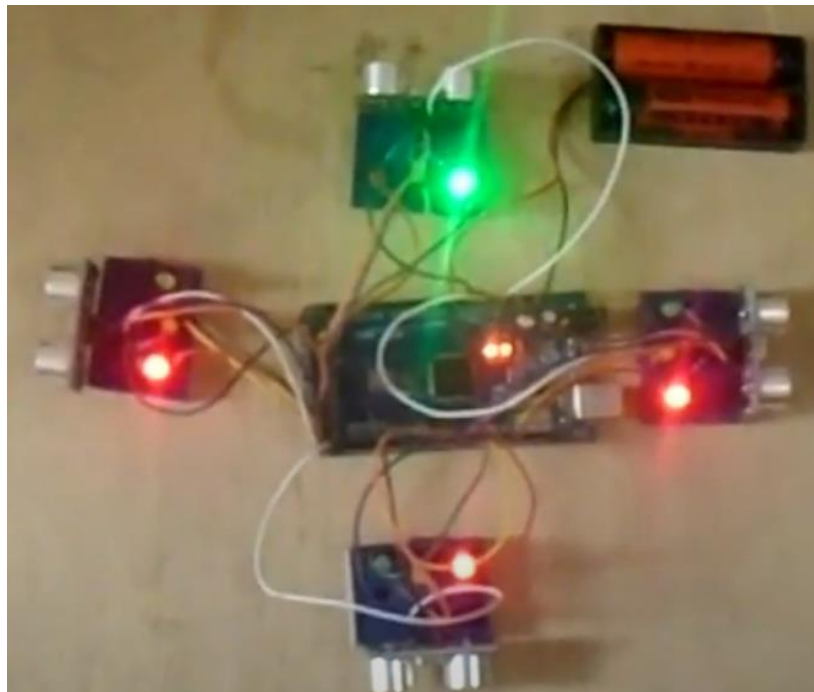


Рисунок 3.18 – Результат роботи пристрою

Таким чином було спроектовано та реалізовано мікроконтролерну систему автоматизованого управління світлофорами на основі аналізу руху автотранспорту, приведено функціональну схему та макет. Запрограмовано дану схему у середовищі Arduino IDE та проаналізовано у EasyEDA та Fritzing.

3.6 Висновки до розділу

Розроблено алгоритм оптимізації транспортних потоків на автострадах. Створено структуру програмного модуля системи виявлення та підрахунку транспорту на автострадах. Описано програмне середовище реалізації програмного забезпечення, зазначено основні функції, які відповідають за реалізацію модулів. Реалізовано програмний засіб на основі алгоритму виявлення та підрахунку транспорту на автострадах. Побудовано модель підсистеми автоматизованого управління світлофорами на основі мереж Петрі. Розроблено структурну схему та модель системи автоматизованого управління світлофорами. Реалізовано програмно-апаратну систему автоматизованого управління світлофорами. Проведено аналіз точності роботи програмного засобу виявлення та підрахунку транспорту на автострадах.

ВИСНОВКИ

1. Досліджено засоби регулювання та методи автоматизації дорожнього руху й зазначено, що до питання регулювання дорожнього руху потрібно знайти комплексний підхід.
2. Проаналізовано засоби реалізації системи управління світлофорами, враховуючи проведений аналіз, було обрано необхідні компоненти для реалізації алгоритму автоматизації
3. Досліджено системи керування дорожнім рухом, виявлено їх основні переваги та недоліки.
4. Проведено аналіз методів оптимізації регулювання дорожнім рухом та евристичних алгоритмів керування транспортними потоками.
5. Розроблено систему оптимізації транспортних потоків
6. На основі розробленої системи розроблено алгоритм оптимізації транспортних потоків на автострадах, який включає в себе комплекс з двох алгоритмів.
7. Розроблено алгоритм виявлення та підрахунку автотранспорту на основі технології комп'ютерного зору та алгоритм автоматизованого управління світлофорами.
8. Розроблено програмне забезпечення виявлення та підрахунку автотранспорту та апаратний засіб автоматизованого управління світлофорами на основі ультразвукового датчика руху.
9. Проведено аналіз точності роботи програмного засобу виявлення та підрахунку автотранспорту. Результатами експериментів підтверджено ефективність програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Девід А. Форсайт, Жан Понсе: Комп'ютерне бачення: сучасний підхід. 2-е видання. 2012, Pearson Education, Prentice Hall 2012.
2. Екземпляр проти семантичної сегментації: які ключові відмінності?: веб-сайт. URL: <https://keymakr.com/blog/instance-vs-semantic-segmentation/> (дата звернення: 15.05.2022).
3. Методи і алгоритми комп'ютерного бачення, що змінюють світогляд: <https://skywell.software/blog/top-6-computer-vision-techniques-and-algorithms/> (дата звернення: 15.09.2022).
4. Методичні вказівки до оформлення курсових, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О.Дубчак / під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 34 с.
5. Методичні рекомендації до виконання дипломної роботи з освітньокваліфікаційного рівня «Магістр». Спеціальність «Комп'ютерна інженерія» / О.М. Березький, Л.О. Дубчак. Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2020. 47 с.
6. Сегментація зображень у 2021 році: архітектури, втрати, набори даних та структури. URL: <https://neptune.ai/blog/image-segmentation-in-2020> (дата звернення: 15.09.2022).
7. Схема виявлення об'єкта. URL: <https://www.fritz.ai/object-detection/#part-basics> (дата звернення: 15.09.2022).
8. Технічні засоби організації руху: веб-сайт. URL: https://msn.khnu.km.ua/pluginfile.php/288552/mod_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%20-%2015.pdf (дата звернення: 23.10.2022).
9. A novel traffic signal control formulation. / Hong Lo. Transportation Research Part B: 1999, 433 – 448 с.

10. A Summary of Vehicle Detection and Surveillance Technologies use in Intelligent Transportation Systems: веб-сайт. URL: <https://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/05pt2.cfm> (дата звернення: 23.10.2022).
11. A dynamic and automatic traffic light control system for solving the road congestion problem: веб-сайт. URL: <https://www.witpress.com/Secure/elibrary/papers/UT06/UT06031FU1.pdf> (дата звернення: 23.10.2022).
12. A multi-band approach to arterial traffic signal optimization. / N.H. Gartner, S.F. Assman, F. Lasaga, and D.L. Hou. Transportation Research Part B: Methodological, 1991, 55–74 с.
13. A multivariable regular approach to traffic responsive network-wide signal control. / C. Diakaki, M. Papageorgiou, and K. Aboudolas. Control Engineering Practice: 2002, 183–195 с.
14. Adaptation in natural and artificial systems. 1975. / J.H. Holland. Ann Arbor, MI: University of Michigan Press and, 1992.
15. Adaptive traffic light control system based on WSN: algorithm optimization and hardware design. / K. Zaatouri, M. H. Jeridi, and T. Ezzedine. International Conference on Software, Telecommunications and Computer Networks (SoftCOM). Split, Croatia, 2018 1–6 с.
16. An IoT based Intelligent Traffic Congestion Control System for Road Crossings: веб-сайт. URL: https://www.researchgate.net/publication/323279531_An_IoT_based_Intelligent_Traffic_Congestion_Control_System_for_Road_Crossings (дата звернення: 23.10.2022).
17. An IoT based intelligent traffic congestion control system for road crossings. / P. Sadhukhan and F. Gazi. International Conference on Communication, Computing and Internet of Things (IC3IoT): Chennai, India, 2018, 371–375 с.
18. Ant colony routing algorithm for freeway networks. / Z. Cong, B. De Schutter, and R. Babuška. Transportation Research Part C: Emerging Technologies, 2013, 1–19 с.

19. Ant system: optimization by a colony of cooperating agents. / M. Dorigo, V. Maniezzo, and A. Coloni. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*: 1996, 29–41 с.
20. Arduino Mega 2560 R3: веб-сайт. URL: <https://amperka.ru/product/arduino-mega-2560> (дата звернення: 23.10.2022).
21. Arterial-based control of traffic flow in urban grid networks. / N.H. Gartner and C. Stamatiadis. *Mathematical and computer modelling*: 2002, 657– 671 с.
22. Automated Real-Time Intelligent Traffic Control System for Smart Cities Using Wireless Sensor Networks: веб-сайт. URL: <https://www.hindawi.com/journals/wcmc/2020/8841893/#B5> (дата звернення: 23.10.2022).
23. Automatic Traffic Management System: веб-сайт. URL: <https://www.parknsecure.com/automatic-traffic-management-system> (дата звернення: 23.10.2022).
24. Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions. / R. Putha, L. Quadrifoglio, and E. Zechman. *Computer-Aided Civil and Infrastructure Engineering*, 27(1): 2012, 14–28 с.
25. Control strategies for over-saturated traffic. / H. Lo and A. H. F. Chow. *ASCE Journal of Transportation Engineering*: 2004, 466–478 с.
26. Cooperative driving: an ant colony system for autonomous intersection management. / J. Wu, A. Abbas-Turki, and A. El Moudni. *Applied Intelligence*, 37(2): 2012, 207–222 с.
27. Design and Performance Analysis of Urban Traffic Control Systems: веб-сайт. URL: <https://core.ac.uk/download/pdf/154747275.pdf> (дата звернення: 23.10.2022).
28. Editorial on wireless networking technologies for smart cities. / J. Lloret, S. H. Ahmed, D. B. Rawat, W. Ejaz, and W. Yu. *Wireless Communications and Mobile Computing*: 2018.

29. Equation of state calculations by fast computing machines. / N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. *The journal of chemical physics*, 21(6): 1953, 1087–1092 c.
30. Future paths for integer programming and links to artificial intelligence. / F. Glover. *Computers & operations research*: 1986, 533–549 c.
31. How Modern Traffic Lights Signals Work: веб-сайт. URL: <https://www.automatesystems.co.uk/traffic-lights-signals-work> (дата звернення: 23.10.2022).
32. Internet of things for smart cities. / A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. *IEEE Internet of Things Journal*: 2014.
33. Intelligent traffic monitoring and guidance system for smart city. / S. Latif, H. Afzaal, and N. A. Zafar. *International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*: Sukkur, Pakistan, 2018, 1–6 c.
34. IoT-based urban traffic-light control: modelling, prototyping and evaluation of MQTT protocol, in 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). / R. Zitouni, J. Petit, A. Djoudi, and L. George. Atlanta, GA, USA, 2019 182–189 c.
35. Network signal setting design with stage sequence optimisation. / S. Memoli, G.E. Cantarella, S. de Luca, and R. Di Pace. *Transportation Research Part B: Methodological*, 100: 2017, 20–42 c.
36. Network traffic signal control with nonconvex alternating direction method of multipliers formulations. / S. Timotheou, C.G. Panayiotou, and M.M. Polycarpou. *Transportation Research Record: Journal of the Transportation Research Board*, 2015, 11–20 c.
37. Optimal freeway ramp metering using the asymmetric cell transmission model. / G. Gomes and R. Horowitz. *Transportation Research Part C*: 2006, 244 – 262 c.
38. Road traffic monitoring system based on mobile devices and bluetooth low energy beacons. / M. Lewandowski, B. Płaczek, M. Bernas, and P. Szymała. *Wireless Communications and Mobile Computing*, 2018.

39. Store-and-forward 184 References based methods for the signal control problem in large-scale congested urban road networks. / K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. *Transportation Research Part C*: 2009, 163–174 с.
40. *Tabu Search* / F. Glover and M. Laguna. Springer, 2013.
41. Traffic congestion evaluation and signal control optimization based on wireless sensor networks: model and algorithms. / W. Zhang, G. Tan, N. Ding, and G. Wang. *Mathematical Problems in Engineering*: 2012.
42. Traffic control. Road traffic control: веб-сайт. URL: <https://www.britannica.com/technology/traffic-control> (дата звернення: 23.10.2022).
43. Traffic signal optimization with greedy randomized tabu search algorithm. / T. Hu and L. Chen. *Journal of Transportation Engineering*, 138(8): 2012, 1040–1050 с.
44. Transport detectors: веб-сайт. URL: <https://www.azd.cz/admin-data/storage/get/446-> (дата звернення: 23.10.2022).
45. Travolution in Ingolstadt: веб-сайт. URL: <http://www.travolution-ingolstadt> (дата звернення: 23.10.2022).
46. Traffic Light Controller Using Arduino: веб-сайт. URL: <https://create.arduino.cc/projecthub/muhammad-aqib> (дата звернення: 23.10.2022).
47. Tendencies of technologies and platforms in smart cities: a state-of-the-art review. / P. Chamoso, A. González-Briones, S. Rodríguez, and J. M. Corchado. *Wireless Communications and Mobile Computing*, 2018.
48. Vehicle Detection and Counting Project — OpenCV Python: веб-сайт. <https://python.plainenglish.io/> (дата звернення: 23.10.2022).
49. What are the Differences Between an Embedded System vs a Microcontroller: веб-сайт. URL: <https://www.totalphase.com/blog/2020/12/differences-between-embedded-system-vs-microcontroller/> (дата звернення: 23.10.2022).
50. What Is a Microcontroller? – Simply Explained: веб-сайт. URL: <https://all3dp.com/2/what-is-a-microcontroller/> (дата звернення: 23.10.2022).