

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра інформаційних систем та технологій

Спеціальність 126 – Інформаційних систем та технологій,
програма “Програмні технології інтернет речей”

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Інформаційна технологія оптимізації міського
дорожнього руху за допомогою IoT»

Студента 2-го курсу групи ІРм-21

Єгор МИРОШНИЧЕНКО

(прізвище, ім'я)

(підпис студента)

Науковий керівник:

Кандидат технічних наук, доцент,

Сергій ПАЛІЙ

(прізвище, ім'я)

(дата)

(підпис)

Попередній захист:

(Висновок: “До захисту в Державній екзаменаційній комісії”)

В.о. завідувача кафедри
інформаційних систем
та технологій

(підпис)

Д.Т.Н., доцент

(науковий ступінь, вчене звання)

Олексій КОЛЕСНИКОВ

(прізвище, ініціали)

(дата)

Київ – 2021

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра: Інформаційних систем та технологій

Освітньо-кваліфікаційний рівень: Магістр

Спеціальність: 126 – Інформаційних систем та технологій

Програма: Програмних технологій інтернет речей

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри,

д.т.н., доцент,

Олексій КОЛЕСНИКОВ

“ _____ ” _____ 2021 року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: *Єгор Максимович МИРОШНИЧЕНКО*

Група: *ІРм-21*

1. Тема кваліфікаційної роботи:

«Інформаційна технологія оптимізації міського дорожнього руху за допомогою IoT»

Затверджена протоколом засідання кафедри ІСТ №16/20 від «09» листопада 2020 р.

Керівник проекту: *Сергій Володимирович ПАЛІЙ*

2. Строк подання студентом готової роботи – «20» травня 2021 р.

3. Цільова установка та вихідні дані до роботи:

Обрати метод фіксації руху транспортних засобів. Розробити IoT систему на базі обраного методу фіксації, що відповідає наступним вимогам:

- *врахувати виявлені недоліки в аналогічних систем і успішно їх нівелювати;*
- *IoT система має бути легко масштабована;*

До функціоналу проектованої системи можна віднести:

- фіксація руху транспортних засобів за рахунок ефекту Холла;
- визначення швидкості і напрямку руху транспортних засобів;
- реалізація алгоритму оптимізації трафіку.

4. Зміст роботи:

- постановка задачі та аналіз рішення;
- аналіз існуючих способів оптимізації автотрафіку;
- опис обраного методу фіксації руху транспортних засобів;
- опис методу оптимізації дорожнього трафіку;
- розробка проекту IoT системи оптимізації трафіку;
- методи та засоби обробки даних з датчиків Холла;
- обґрунтування обраних технологій для створення програмного продукту.

5. Перелік графічних матеріалів (слайдів):

- схема електрична-принципова системи моніторингу дорожнього трафіку на базі датчиків Холла;
- схема вимірювання швидкості транспортного засобу;
- архітектурне рішення системи IoT рішень для оптимізації трафіку;
- макет прототипу системи виміру збурень магнітного поля;
- алгоритм роботи та структурна схема системи оптимізації;
- графічний інтерфейс системи в LabVIEW.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	Дати виконання роботи за планом	Виконання роботи за планом
1	Вивчення літературних джерел з оптимізації дорожнього руху	05.02.2021 – 12.02.2021	виконано

2	Збір і вивчення матеріалів на базі ТОВ «Інформаційні технології України»	13.02.2021 – 20.02.2021	виконано
3	Складання розгорнутого плану кваліфікаційної роботи	20.02.2021 – 25.02.2021	виконано
4	Підготовка розділу «ФОРМУВАННЯ ТЕХНІЧНОГО ЗАВДАННЯ»	26.02.2021 – 07.03.2021	виконано
5	Підготовка розділу «РОЗРОБКА ПРОЕКТУ ІоТ-РІШЕННЯ ТА МЕТОДІВ ОБРОБКИ ДАНИХ В ІоТ»	08.03.2021 – 14.03.2021	виконано
5	Підготовка розділу «РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ІоТ-РІШЕННЯ»	15.03.2021 – 25.03.2021	виконано
6	Підготовка розділу «ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСНИХ РІШЕНЬ ТА ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ»	26.03.2021 – 12.04.2021	виконано
7	Оформлення кваліфікаційної роботи	26.03.2021 – 19.04.2021	виконано
8	Передача кваліфікаційної роботи рецензенту для рецензування	20.04.2021 – 30.04.2021	виконано
9	Передача кваліфікаційної роботи науковому керівникові	20.04.2021 – 30.04.2021	виконано
10	Попередній захист кваліфікаційної роботи	20.04.2021	виконано
11	Захист роботи	26.05.2021	

Дата видачі завдання «10» листопада 2021 р.

Керівник роботи: *доцент кафедри інформаційних систем та технологій
Сергій ПАЛІЙ*

(підпис)

Завдання прийняв до виконання:
студент групи ІРм-21 Єгор МИРОШНИЧЕНКО

(підпис)

АНОТАЦІЯ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра Інформаційних систем та технологій
Освітня програма «Програмні технології інтернет речей»

Кваліфікаційна робота магістра Єгора МИРОШНИЧЕНКА

Тема роботи: «Інформаційна технологія оптимізації міського дорожнього руху за допомогою IoT».

Мета кваліфікаційної роботи магістра – стало створення інформаційної технології оптимізації міського дорожнього руху за допомогою засобів IoT на базі збору і опрацювання даних зчитаних з чутливих сенсорів магнітного поля.

Об’єкт дослідження – стала інформаційно-апаратна система для визначення інтенсивності руху транспортних засобів. Програмно-апаратна реалізація належить до області систем оптимізації дорожнього руху. І задумана для оптимізацію дорожнього трафіку мегаполісів.

Предмет дослідження – спосіб фіксації руху автотранспорту і розробка відповідного програмного забезпечення для опрацювання зчитаних показників.

Методи дослідження – використані теоретичні методи(класифікація, синтез, аналіз) емпіричні (експеримент, узагальнення), аналіз отриманих результатів шляхом статистичної обробки. Розроблено програмний код для мікроконтролеру у Arduino IDE, розроблено графічний інтерфейс для візуалізації результатів вимірів у LabVIEW.

Практичне значення одержаних результатів. Дана кваліфікаційна робота магістра має високе практичне, це пов’язано з тим, що вдалося експериментально довести працездатність обраного методу фіксації руху на базі давачів Холла. Окрім цього запропонована система нівелює всі проблеми виявлені у аналогічних розробок і дозволяє виконувати легке і швидке

масштабування за рахунок IoT. Одержані результати рекомендовано впровадити на виробництво.

Апробація результатів роботи була виконана на двох конференціях:

1. Myroshnychenko Y. Road traffic optimization by IoT [Електронний ресурс] / Y. Myroshnychenko, S. Paliy // Information technology & Interactions (Satellite)». – 2020. – Режим доступу до ресурсу: http://iti.fit.univ.kiev.ua/wp-content/uploads/ITI_Satellite_2020_Conference-Proceedings.pdf. [45]

2. Мирошніченко Є.М. Оптимізація руху автотранспорту з використанням розподіленої системи збору даних на базі датчиків Холла : наукова публікація. XVII міжнародна науково-технічна конференція «Проблеми інформатизації», 2021 р. [46]

Кваліфікаційна робота магістра складається зі змісту, вступу, основної частини, яка включає чотири розділи, висновків та списку використаних джерел та додатків. Всього 99 сторінок.

КЛЮЧОВІ СЛОВА: давачі Холла, інформаційна IoT технологія, мікроконтролер, моніторинг руху транспортних засобів, оптимізація руху транспортних засобів.

ABSTRACT

TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV

Faculty of Information Technologies

Department of Information Systems and Technologies

Educational Program "Software Technologies of the Internet of Things"

Qualification work of master Yehor MYROSHNYCHENKO.

Work topic: «Information technology of city traffic optimization by IoT».

The purpose of the master's qualification work was to create an information technology for optimizing urban traffic with the help of IoT on the basis of collecting and processing data read from sensitive magnetic field sensors.

The object of research - a permanent information and hardware system for determining the intensity of traffic. Software and hardware implementation belongs to the field of traffic optimization systems. And designed to optimize road traffic in cities.

The subject of research - a method of recording the movement of vehicles and the development of appropriate software for processing the readings.

Research methods - used theoretical methods (classification, synthesis, analysis) empirical (experiment, generalization), analysis of the results by statistical processing. Developed software code for the microcontroller in the Arduino IDE, developed a graphical interface for visualization of measurement results in LabVIEW.

The practical significance of the obtained results. This master's qualification work is highly practical, this is due to the fact that it was possible to experimentally prove the effectiveness of the chosen method of motion detection based on Hall sensors. In addition, the proposed system eliminates all the problems identified in similar developments and allows you to perform easy and fast scaling due to IoT. It is recommended to implement the obtained results in production.

Approbation of the results of the work was performed at two conferences:

1. Myroshnychenko Y. Road traffic optimization by IoT [Електронний pecypc] / Y. Myroshnychenko, S. Paliy // Information technology & Interactions

(Satellite)». – 2020. – Режим доступу до ресурсу: http://iti.fit.univ.kiev.ua/wp-content/uploads/ITI_Satellite_2020_Conference-Proceedings.pdf. [45]

2. Мирошниченко Є.М. Оптимізація руху автотранспорту з використанням розподіленої системи збору даних на базі датчиків Холла : наукова публікація. XVII міжнародна науково-технічна конференція «Проблеми інформатизації», 2021 р. [46]

The master's qualification work consists of the content, introduction, main part, which includes four sections, conclusions and a list of used sources and appendices. Total 99 pages.

KEY WORDS: Hall sensors, IoT information technology, microcontroller, vehicle traffic monitoring, vehicle traffic optimization.

ЗМІСТ

ВСТУП	11
РОЗДІЛ 1. ФОРМУВАННЯ ТЕХНІЧНОГО ЗАВДАННЯ.....	16
1.1 Аналіз існуючих способів моніторингу руху транспортних засобів	16
1.2 Аналіз існуючих способів оптимізації автотрафіку.....	18
1.3 Опис обраного методу фіксації руху транспортних засобів.....	22
1.4 Опис методу оптимізації дорожнього трафіку	26
1.5 Висновки до першого розділу	30
РОЗДІЛ 2. РОЗРОБКА ПРОЕКТУ ІоТ-РІШЕННЯ ТА МЕТОДІВ ОБРОБКИ ДАНИХ В ІоТ	31
2.1. Розробка проекту ІоТ системи оптимізації трафіку.....	31
2.2. Методи та засоби обробки даних з давачів Холла.....	33
2.3. Обґрунтування обраних технологій для створення інформаційної системи	36
2.3.1 Обґрунтування вибору Arduino Pro Mini.....	36
2.3.2 Обґрунтування вибору середовища для візуалізації експериментальних даних — LabVIEW	37
2.4. Обґрунтування обраного методу оптимізації синхронізації дорожнього сигналу	38
2.4.1 Вибір та обґрунтування алгоритму оптимізації	42
2.5. Висновки до другого розділу	46
РОЗДІЛ 3. РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ІоТ-РІШЕННЯ	48
3.1. Архітектура мікропроцесорної системи	48
3.2. Комунікаційні технології та системи використані у проекті.....	52
3.3. Проміжне програмне забезпечення	54
3.4. Архітектура проекту оптимізації трафіку на базі ІоТ-рішення	57

3.5. Аналіз захищеності інформації та протоколів для її кіберзахисту	59
3.6. Висновки до третього розділу	64
РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСНИХ РІШЕНЬ ТА ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРУ	66
4.1. Реалізація програмної частини проекту в Arduino IDE	66
4.2. Реалізація візуалізації проекту в LabVIEW	72
4.3. Висновки до четвертого розділу	75
ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	79
Додаток А.....	85
Додаток Б	90

ВСТУП

Актуальність теми. Злагоджене функціонування транспортних магістралей є однією із основних умов ефективності сучасної економіки. За даними міністерства інфраструктури України станом на 2018 рік в Україні зареєстровано більше 9,2 млн. транспортних засобів. У 2018 році кількість авто, що припадала на тисячу громадян, становила 219 автомобілів, в цей же час у 2016 році на тисячу громадян припадало всього 202 автомобілі. Оперуючи цими даними проглядається тенденція до збільшення кількості транспортних засобів в Україні [1].

Судячи з цього, питання оптимізації трафіку міст є дуже важливим, для цього необхідно ефективно оцінювати проблемні ділянки доріг, оперативно отримувати актуальну дорожньо-транспортну інформацію, і як наслідок ефективно керувати автомобільним рухом у містах (керування тривалістю світлофорних режимів та своєчасного вмикання інформаційних табло) [2].

Сучасні системи моніторингу дорожнього руху мають ряд недоліків які впливають на якість фіксації інтенсивності трафіку і подальшої обробки даних. Зазвичай це проявляється при паркуванні автомобілів на смугах руху, чи при перекритті транспортним засобом сенсора / візуального спостерігача (зазвичай при багаторядному русі).

Проаналізувавши сучасні системи моніторингу було виявлено, що «вузьким» місцем більшості систем є безпосередньо методи/давачі які фіксують рух [3]. Виходячи з цього, головною задачею при проектуванні нової системи моніторингу дорожнього руху став вибір методу фіксації руху автомобілів який зможе нівелювати виявлені проблеми.

У зв'язку із збільшення автотрафіку по всьому світу, загострюється питання боротьби із заторами. Щоб заглибитись у вирішення цього питання необхідно розібратися із самим поняттям затор.

Затори на дорогах - це стан руху транспорту, який у зв'язку із збільшенням кількості дорожнього транспорту характеризується повільними швидкостями, більш тривалим часом поїздки та збільшеними чергами транспортних засобів на певну (деколи досить протяжну) ділянку дороги. Коли трафік настільки великий, що взаємодія між транспортними засобами уповільнює швидкість потоку трафіку, це призводить до певної завантаженості. У той час як затори можливі для будь-якого виду транспорту, дана проблема буде розглянута зі сторони автомобільних заторів на дорогах загального користування.

У міру того, як пропускна можливість наближається до пропускної спроможності дороги, починається формування затору. Коли автотрафік повністю зупиняється на певний період часу, саме це безпосередньо і називають затором. Затори на автомагістралях можуть привести до багатьох негативних наслідків. До цих наслідків можна віднести [3]:

- дорожня лють («road rage», термін виник у США в 1987–1988 роках) — агресивна чи зла поведінка з боку водія автомобіля або іншого транспортного засобу;
- різноманітні економічні збитки (збільшення часу в дорозі, запізнення на роботу);
- зниження пропускної спроможності дороги;
- унеможливлення роботи екстрених служб через блокування доріг;
- збільшення забруднення оточуючої середовища за рахунок збільшення витрат палива;
- збільшення шумового фону міст;
- збільшення аварійних ситуацій.

Математично перевантаженість магістралей зазвичай розглядається як кількість транспортних засобів, які проходять через ділянку дороги за одиницю часу.

Основним чинником заторів є підвищення кількості авто на дорогах, що перевищує їх пропускну спроможність. Збільшення кількості заторів може залежати від:

- сезонних переїздів «на море / у гори»;
- щоденних поїздок на роботу в «час пік»;
- об'їзд водіяма перекритих ділянок доріг в наслідок ремонтних робіт, проведення масових заходів, дорожньо-транспортних пригод (ДТП).

Ці чинники досить непередбачувані за своєю інтенсивністю і часом виникнення. В свою чергу пропускну здатність ділянок доріг також може змінюватись від постійних і випадкових факторів.

Також до причини, що сприяють зменшенню пропускну можливості доріг:

- порушення ПДД;
- створення ДТП;
- ремонтні роботи або прибирання на дорогах в «час пік»;
- несприятливі погодні умови, або техногенні проблеми (смог від пожеж);
- нерегульовані перехрестя з рухом «по колу»;
- паркування авто безпосередньо на полосі руху;
- неправильна конструкція доріг;
- світлофори з недостатньою тривалістю режимів роботи яка не враховує інтенсивність трафіку.

Саме тому, було вирішено реалізувати інформаційно-апаратне рішення для оптимізації міського руху на базі засобів IoT для керування рухом у мегаполісах, керування тривалістю світлофорних «коридорів».

Метою магістерської роботи стало створення інформаційної технології оптимізації міського дорожнього руху за допомогою засобів IoT на базі збору і опрацювання даних зчитаних з чутливих сенсорів магнітного поля.

Об'єктом магістерської роботи стала інформаційно-апаратна система для визначення інтенсивності руху транспортних засобів. Програмно-апаратна реалізація належить до області систем оптимізації дорожнього руху. І задумана для оптимізацію дорожнього трафіку мегаполісів.

Предмет роботи є спосіб фіксації руху автотранспорту і розробка відповідного програмного забезпечення для опрацювання зчитаних показників.

При написанні магістерського проекту були використані наступні **методи дослідження**: теоретичні (класифікація, синтез, аналіз) емпіричні (експеримент, узагальнення), аналіз отриманих результатів шляхом статистичної обробки.

Отримані результати мають високе **практичне значення**, це пов'язано з тим, що вдалося експериментально довести працездатність обраного методу фіксації руху на базі давачів Холла [4]. Окрім цього запропонована система нівелює всі проблеми виявлені у аналогічних розробок і дозволяє виконувати легке і швидке масштабування за рахунок IoT.

Апробація результатів роботи була виконана на двох конференціях:

1. «Road traffic optimization by IoT» на конференцію «Information technology & Interactions (Satellite)» в 2020 році;
2. «Оптимізація руху автотранспорту з використанням розподіленої системи збору даних на базі датчиків Холла» на сімнадцяту міжнародну науково-технічну конференцію «ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ» в 2021 році.

В магістерській роботі запропоновано технологію оптимізації міського дорожнього руху засновану на давачах Холла для фіксації даних про рух автомобілів і засобах IoT для об'єднання поодиноких систем фіксацій у єдину

«екосистему» міста. Для перевірки працездатності системи зібрано апаратну частину експериментального зразку модулю зчитування і обробки даних з давачів Холла. Дослідно перевірено доцільність використання магнітно-резонансного ефекту для фіксації і подальшої оптимізації руху транспортних засобів. Експериментальні дані візуалізовано на базі програмного застосунку LabVIEW, що в повній мірі довело працездатність системи.

РОЗДІЛ 1. ФОРМУВАННЯ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз існуючих способів моніторингу руху транспортних засобів

Однією з проблемних сторін життя в мегаполісі є перенасичений дорожній рух. Особливо великою неприємністю стають вуличні затори. Для усіх міських жителів ситуація із заторами ще надовго залишатиметься значною проблемою. Та вже сьогодні необхідно прикладати максимум зусиль для боротьби з даною проблемою.

Першим етапом для вирішення проблеми із заторами в Українських містах має стати введення автоматизованих системи управління дорожнім трафіком (АСУДТ). АСУДТ є важливою складовою інтелектуальних систем керування транспортними потоками, і виконує функції керування та інформування, серед яких є:

- керування і оптимізація потоків транспортних засобів;
- забезпечення транспортних магістралей актуальною інформацією;
- підвищення безпеки на небезпечних ділянках доріг;
- допомога при керуванні в особливих ситуаціях [5].

В цілому підсистеми АСУДТ можуть бути представлені як сукупність давачів дорожньої телематики та мікроконтролерів, підключених до мережі обміну даними, з організацією туманних обчислень на місцях зчитування інформації і глобальних дата центрів.

До основних складових будь-якої системи моніторингу інтенсивності руху транспортних засобів відносяться давачі дорожньо-транспортних показників і елементи візуального спостереження. Від апаратних можливостей давачів та достовірності знятих показників залежить якість роботи всієї системи моніторингу. Більшість систем моніторингу трафіку ґрунтуються на застосуванні зональних давачів — сукупність давача та мікропроцесорного модулю обробки зчитаних даних [6].

Існуючі системи моніторингу погано справляються із своїми задачами в умовах перевантаження транспортних магістралей, паркування автівок на смугах руху автотранспорту, при перекритті транспортним засобом давача, що відповідає за фіксацію руху авто та при русі транспорту поміж смуг руху. Зазвичай, це призводить до помилкового переходу системи моніторингу трафіку до стану «затор», та подальших проблем що виникають в наслідок помилкового спрацювання системи (спричинення заторів, неправильне керування інформаційними табло і т.д.).

Саме тому впровадження нових методологій зчитування показників із давачів, що виконують моніторинг автотрафіку на базі мікропроцесорних рішень, а також розробка більш досконалого програмного забезпечення, під нові методи, мають вирішити актуальну технічну проблему з оптимізації руху транспортних засобів у містах по усьому світу. Це надає змогу ефективно управляти дорожнім рухом мегаполісів і завчасно попереджати водіїв про можливі небезпеки.

Для аналізу і оцінки стану об'єкта дослідження було порівняно декілька різних способів моніторингу руху транспортних засобів [7]:

1. За допомогою оптичних датчиків:

- системи з оптичними давачами – перетин оптичного променю (при встановлені на дороги із кількістю смуг більшою за три унеможливується кількісна оцінка, та зникає можливість зафіксувати напрямок руху і розрахувати швидкість транспортного засобу);
- схема з відбиттям променю від днища транспортного засобу унеможливується у випадку забруднення, або наявності масляних шарів на днищі;
- проблема із захистом оптичних давачів від забруднення (вони завжди повинні бути відкриті – особливість роботі системи).

2. За допомогою ультразвукових датчиків [8]:
 - системи з ультразвуковими давачами — перетин ультразвукового променя (при встановленні на дороги із кількістю смуг більшою за три унеможливується кількісна оцінка, та зникає можливість зафіксувати напрямок руху і розрахувати швидкість транспортного засобу);
 - проблема із захистом ультразвукових давачів від забруднення (вони завжди повинні бути відкриті – особливість роботи системи).
3. За допомогою індуктивних датчиків:
 - магнітні ворота (дорого коштують, погано захищені від вандалізму).

1.2 Аналіз існуючих способів оптимізації автотрафіку

Перед реалізацією системи оптимізації руху транспортних засобів, був проведений пошук аналогів що працюють з використанням схожих рішень, і які б задовольнили вимоги висунуті до системи.

Першим патентом, який було розглянуто, що використовує у своїй роботі ефект Холла став «Railway wheel sensor employing hall effect devices and corresponding method», авторів Glen Appleby, Stephen W. Brown, Kostas Parazoglou, у 2005 році [9]. Патент зареєстровано в базі Європейського патентного офісу під номером EP1899209B1. Цей патент описує метод детектування наявності рейкового транспортного. Патент не несе прикладного функціоналу в рамках моєї магістерської роботи, і розглядається виключно для ознайомлення з інтерпретаціями методів використання ефекту Холла.

Наступним патентом стала «System for automated control of the driver's compliance with traffic rules and determination of the fact of an accident» [10]. Цей патент зареєстровано від 2009 року. Патентне рішення базується на використанні багатофункціональної бортової апаратури, систем збору і обробки даних. Запропоноване патентне рішення має певний ряд недоліків:

- 1) відсутність автоматичної обробки зчитаних даних;
- 2) низький темп зчитування параметрів руху за допомогою GPS трекера.

Останнім розглянутим патентним рішенням став «Спосіб визначення швидкості руху і координат транспортних засобів з подальшою їх ідентифікацією та автоматичною реєстрацією порушень правил дорожнього руху і пристрій для його здійснення». Патент зареєстровано у WIPO за номером WO2011096840A1, у 2010 році [11]. Дане рішення, базується на випроміненні в напрямку руху автомобіля електромагнітного променя, і фіксації відбитих імпульсів. Швидкість руху автомобілів і відстань до них обчислюється шляхом порівняння значень випромінених і прийнятих імпульсів. В разі реєстрації перевищення швидкості, відеокамера ідентифікує транспортний засіб і фіксує порушення швидкісного режиму в автоматичному режимі. Дане рішення має недолік у вигляді помилкового визначення «автомобіля-порушника» у разі багаторядного руху.

Дослідивши різні патентні рішення та запропоновані реалізації, було прийняте рішення створити власну систему засновану на ефекті Холла, з принципово новою архітектурою та реалізацією.

Пропоную розглянути найбільш розвинену із усіх існуючих на даний момент систем, що наразі запропонована і реалізована компанією «Rapid Flow Technologies». Система називається «Surtrac» - це інтелектуальна система управління сигналами дорожнього руху, заснована на стику технологій робототехніки та штучного інтелекту [12]. Surtrac трактує проблему оптимізацію трафіку наступним чином: «As a single machine scheduling problem». Тобто врахувавши кожен окрему автівку у транспортному потоці і розрахувавши загальне завантаження система передає данні до сусідніх підсистем, щоб оптимізувати трафік на наступних ділянках дороги.

Алгоритм роботи системи:

1. Перше, що робить система Surtrac, як і будь-яка роботизована система — збирає данні із свого оточення про те, що відбувається в реальному часі у оточуючому середовищі. Цю інформацію система отримує із інтеграції із навколишнім програмним забезпеченням за допомогою API інтеграцій із існуючою інфраструктурою, яка може включати камери, радари або навіть індукційні ворота.

2. Surtrac обробляє цю інформацію, а потім щосекунди за допомогою програмного забезпечення створює план оптимізації руху автотрафіку через перехрестя.

3. Наступним кроком Surtrac надсилає команди контролеру для керування системами інформування водіїв / зміненню світлофорних тактів.

4. Нарешті, Surtrac передає інформацію про зчитані данні по завантаженості ділянки дороги:

- до сусідніх перехресть, щоб вони могли включити цю інформацію у свої плани по оптимізації, що дозволяє виконувати як автономне керування трафіком на ділянках доріг, так і виконувати оптимізацію на рівні мережі Surtrac, в залежності від умов дорожнього руху.

- до розумних транспортних засобів, пасажирів, пішоходів або будь-якого іншого цифрового пристрою або системи, які можуть використовувати цю інформацію.

Система Surtrac розроблена спеціально для міських дорожніх мереж, де рух транспорту є складним, динамічним та мультимодальним. Система адаптується до несподіваних заторів або до зменшення пропускної спроможності ділянок дороги. Surtrac враховує перекриттям смуг руху, паркуванням транспортних засобів на дорожньому полотні та іншими джерелами невизначеності в режимі реального часу.

На наступному рисунку зображена схема роботи системи (рис. 1.1):

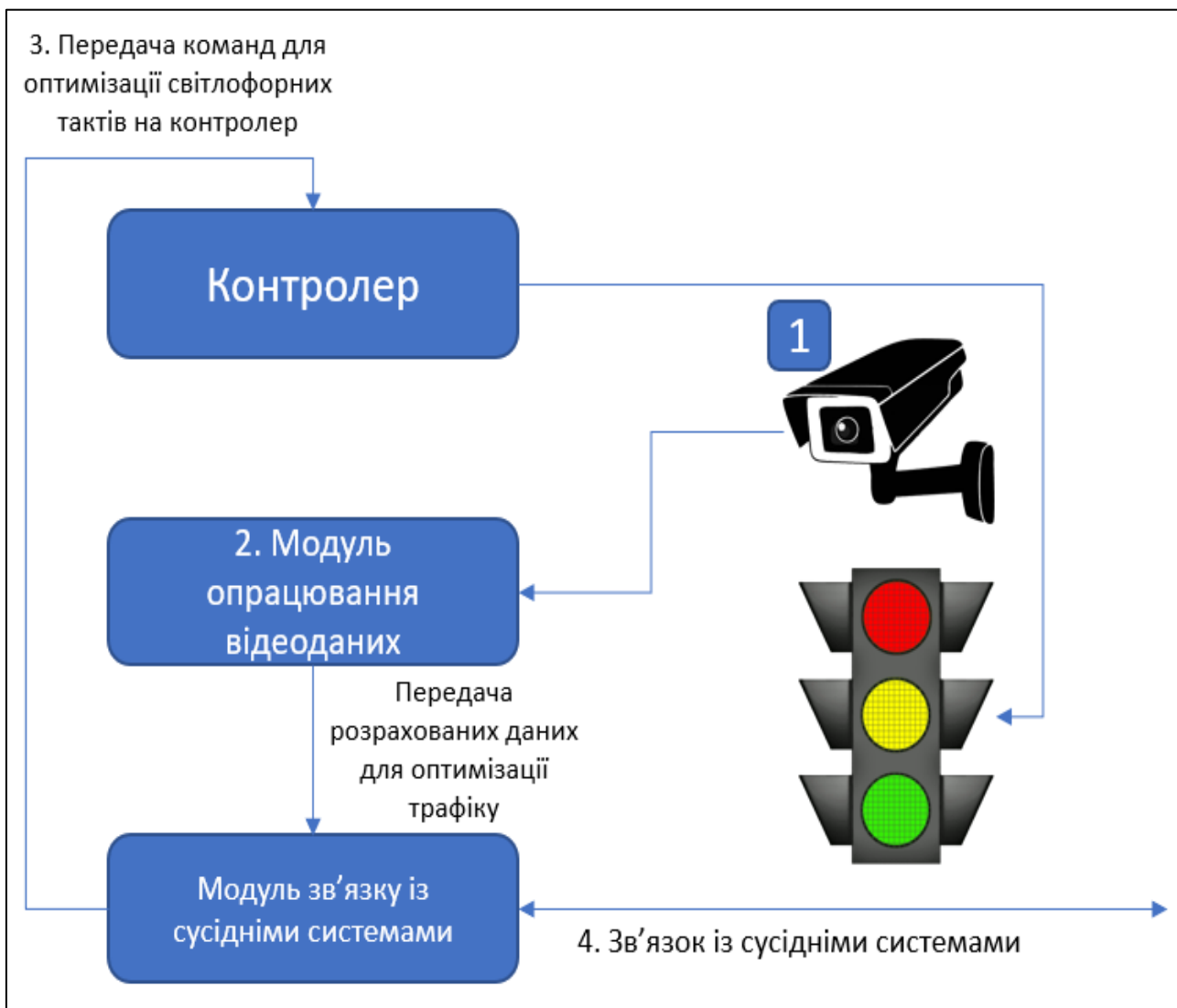


Рисунок 1.1. Система оптимізації трафіку від Surtrac.

За рахунок об'єднання поодиноких систем у єдину екосистему міста інформація про завантаженість доріг з перехресть у реальному часі дозволяє Surtrac уникати направлення транспортних засобів у вже перенасичене транспортне «русло».

Та незважаючи на гарно продуману інфраструктуру запропоновану компанією «Rapid Flow Technologies» у цієї системи є суттєвий недолік — саме технологія завдяки якій компанія фіксує інформацію про інтенсивність дорожнього руху. Для цього вони використовують, у більшій мірі камери відеоспостереження вартість яких є досить високою. До допоміжних давачів

інформації стану дорожнього трафіку, які використовую компанія можна віднести радари і індукційні ворота. А як вже зазначалося у вступі, то саме ці давачі погано справляються із перекриттям смуг руху, паркуванням авто на смугах руху і забрудненням [13].

Саме тому мною було прийняте рішення розробити систему з схожою адаптивністю до масштабування і логікою роботи, але використати для фіксації руху автотранспорту давачі Холла.

1.3 Опис обраного методу фіксації руху транспортних засобів

Приймаючи до уваги існування великої кількості досить різних підходів до фіксації дорожнього руху:

- системи з відео фіксацією;
- системи з перетином оптичного чи ультразвукового променя;
- системи з використання індукційних воріт.

На даний момент не існує аналогів системи функціонування якої базується на основі давачів Холла, запропонованої в магістерській роботі. Оригінальність і новаторство запропонованої системи підтверджені результатами патентного пошуку.

Кожний автомобіль має у своєму складі металеві запчастини і, при своєму русі, транспортний засіб у будь-якому випадку збуджує магнітне поле Землі (рис. 1.2) [14].

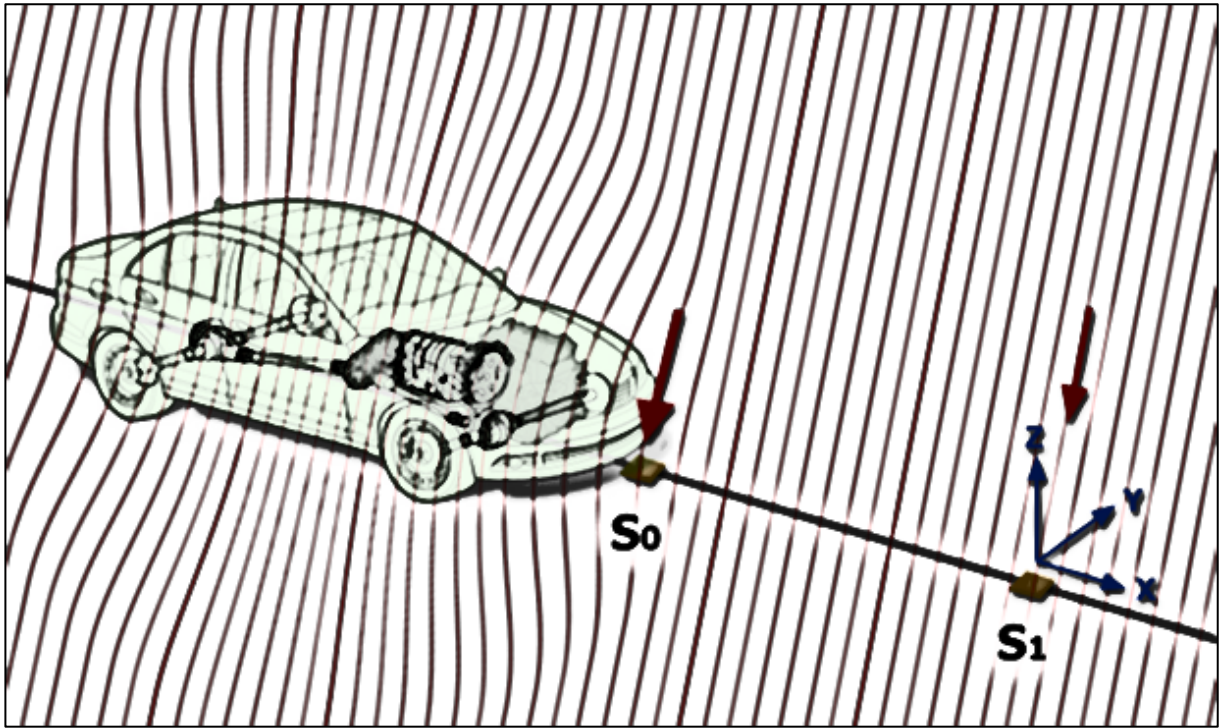


Рисунок 1.2. Вплив ферромагнітних елементів авто на магнітне поле.

Для оптимізації автотрафіку міст була поставлена задача створити IoT систему моніторингу дорожнього трафіку на основі чутливих датчиків фіксації магнітного поля. Пристрій містить мікропроцесор-контролер — Arduino, два датчі Холла (рис. 1.3).

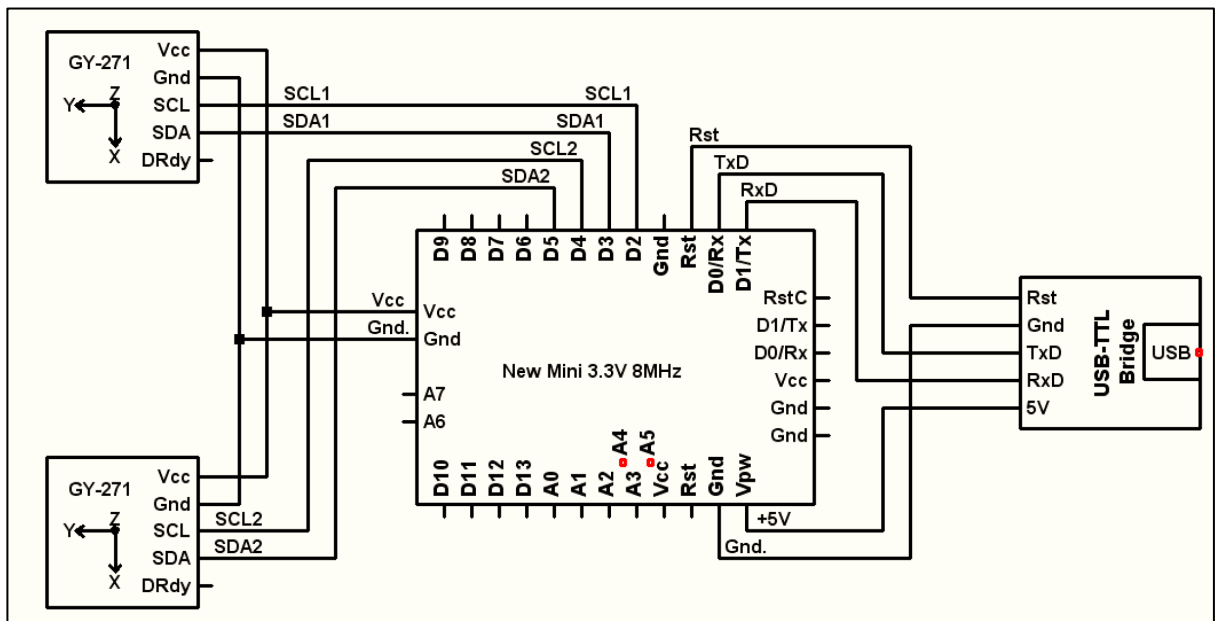


Рисунок 1.3. Схема електрична-принципова системи моніторингу дорожнього трафіку на базі датчиків Холла.

Проаналізувавши зміну магнітного поля Землі спричиненого рухом транспортного засобу зі сталевими і залізними деталями над сенсорами магнітного поля і вимірюючи затримку між моментами спрацьовування двох сенсорів (рис. 1.4) можна визначити швидкість руху транспортного засобу за формулою 1.1:

$$V = \frac{\Delta L}{\Delta t}, \quad (1.1)$$

де ΔL — дистанція між давачами S_0 і S_1 ;

Δt — різниця в моментах спрацьовування давачів.

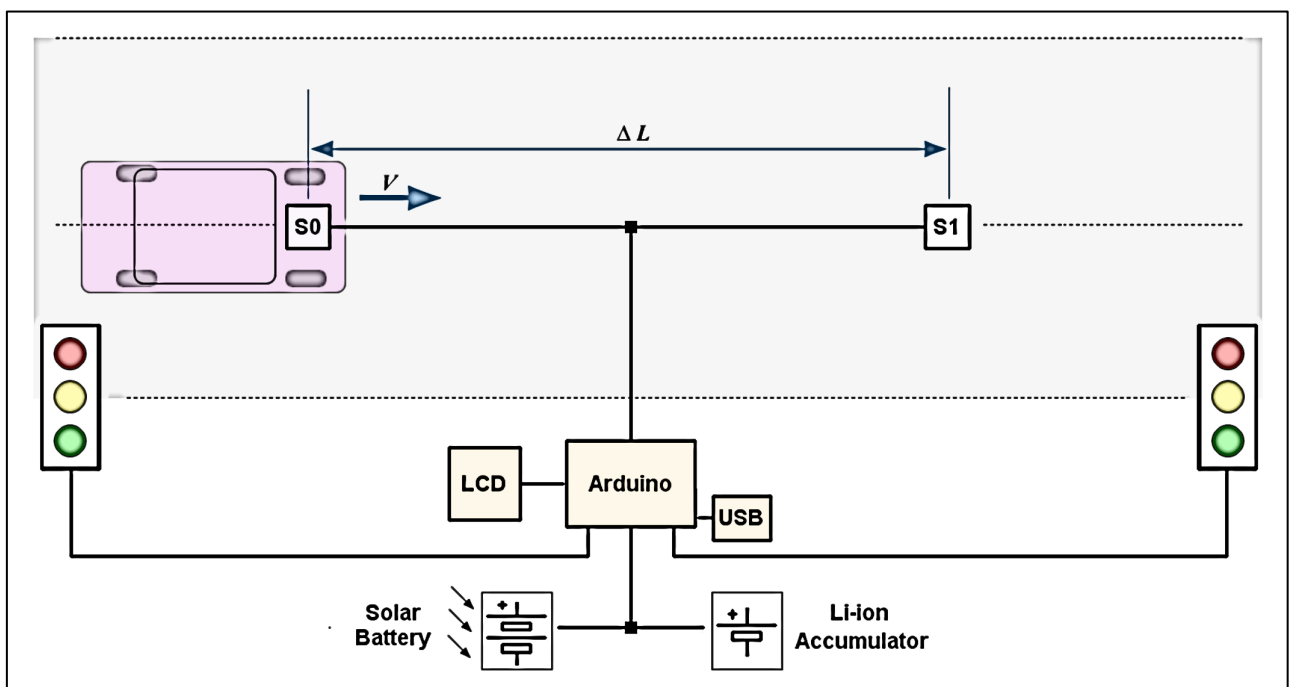


Рисунок 1.4. Схема вимірювання швидкості транспортного засобу.

Метод розрахунку зсуву об'єкта можна описати таким чином:

- ініціалізація цифрового інтерфейсу для двох давачів;
- ініціалізація кожного з давачів на вимірювання. Для цього підбираються наступні параметри: метод усереднення, частота опитування давачів, значення чутливості;
- вимірювання магнітного поля по координатах X , Y , Z кожного з давачів;

- ініціалізація початкових значень для процедури усереднення і подальшого вилучення постійної складової магнітного поля Землі;

Після цього в нескінченному циклі мікропроцесор виконує наступну послідовність дій:

- розраховується значення магнітного поля з даних отриманих з двох датчиків;
- для кожного датчика по координатам X, Y, Z обчислюється значення відхилення від поточного середнього значення;
- обчислюється модуль відхилення по координатам X, Y, Z для обох датчиків;
- серед відхилень по всім координатам обчислюється максимум для кожного датчика. Цей максимум характеризує величину збурення для даного датчика.

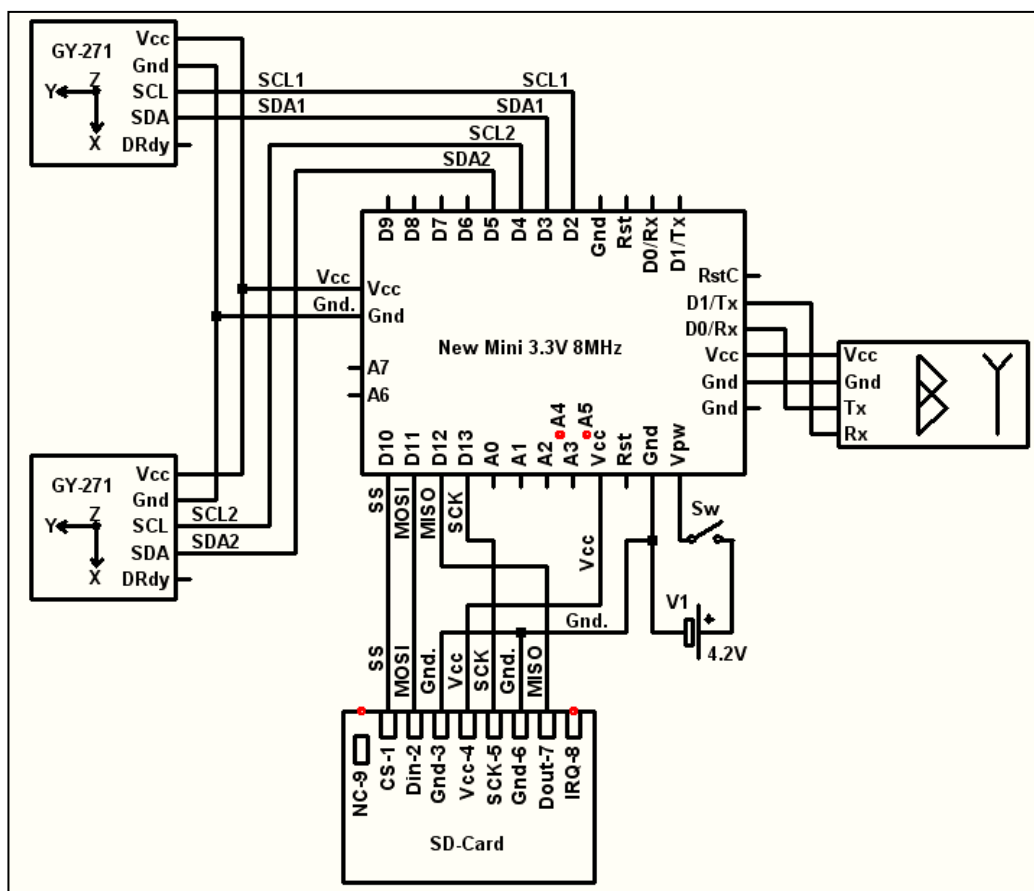


Рисунок 1.5. Система фіксації руху на базі розподіленої системи збору даних.

Величина вимірюного обурення по кожному сенсору передається для подальшої обробки. При реалізації методу моніторингу, описаного вище, важливу роль відіграє саме метод зберігання і обробки інформації. Для реалізації системи оптимізації дорожнього трафіку в місті необхідно поєднати кожен окремий контролер (рис. 1.5) в єдину екосистему IoT, яка могла оперативнo отримувати і обробляти дані, і відповідно, на базі отриманих розрахунків управляти світлофорними тактами в містах. Всі ці процеси повинні виконуватися дуже швидко, тому що інформація зчитана з контролерів актуальна тільки незначний проміжок часу. Більш того, система повинна працювати злагоджено, так як це безпосередньо впливає на збереження людського життя. Всі ці вимоги повною мірою покриваються можливостями IoT!

1.4 Опис методу оптимізації дорожнього трафіку

В основу обраного методу оптимізації трафіку міст було покладено принцип зчитування зміни постійної складової магнітного поля Землі за рахунок наявності у транспортних засобах феромагнітних частин, використовуючи чутливі датчики магнітного поля – датчики Холла.

Для коректного функціонування запропонованої системи необхідна наявність двох давачів Холла встановлених під дорожнє полотно на кожній смузі руху розташованих на фіксованій відстані, це зумовлено логікою фіксації руху.

Вирахувавши затримку між моментами спрацьовування двох давачів можна визначити швидкість руху транспортного засобу. Реєструючи профіль зміни збурення в часі над одним давачем, і знаючи швидкість руху транспортного засобу, можна визначити його довжину. Знаючи який із давачів спрацював раніше можна визначати напрямок руху транспортного засобу.

Крім того, можна побудувати профіль збурення по довжині об'єкта. Знаючи довжину і профіль збурення, та маючи бібліотеку образів профілів для різних типів об'єктів, можна розпізнавати тип транспортного засобу.

У поданій схемі системи реєстрації руху (рис. 1.6) розглядається можливість передачі даних отриманих системою за рахунок підключенням до навколишнього середовища на базі стандарту IEEE 802.15.4.

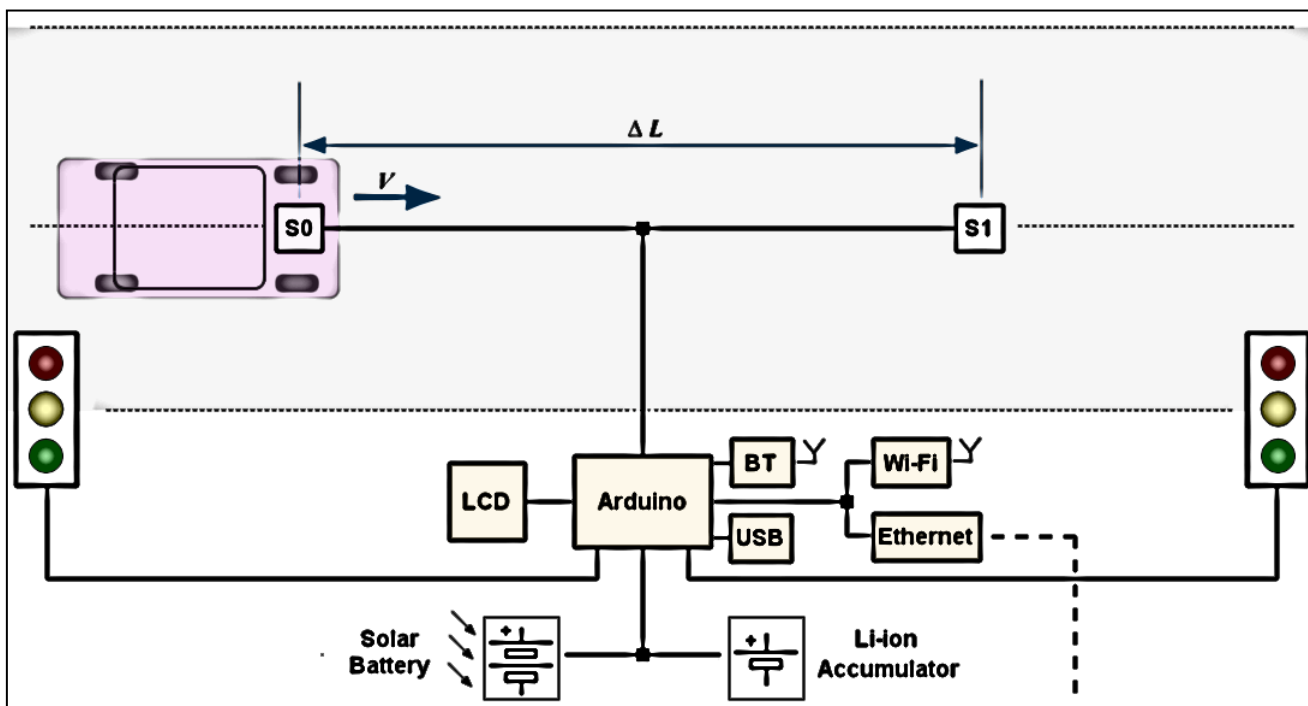


Рисунок 1.6. Система реєстрація руху з урахуванням масштабування.

Стандарт IEEE 802.15.4 представляє собою суттєвий відступ від «більших і швидших» стандартів, які впроваджує організація IEEE 802: замість більш високої швидкості передачі даних та більшої функціональності цей стандарт вирішує питання контрольних та сенсорних мереж з низьким обсягом даних, які довгий час існували без глобальної стандартизації. В основі цього стандарту - RF і PHYsical рівень з трансивером Phase-Shift-Key (PSK), здатним розвивати швидкість бездротового зв'язку до 250 кілобіт в секунду (кбіт/с), що працює на підмножині з 27 доступних радіоканалів у певних неліцензійних (залежно від географічного регіону) смугах 800, 900 та 2400 МГц. Проблеми з недостатньою кількістю радіочастот каналу вирішуються за допомогою спектру розширення прямої послідовності (DSSS). Використовуються два способи доступу до каналу:

- багаторазовий доступ Carrier Squelch (CSMA) із запобіганням зіткнення (CA);

- багаторазовий доступ до часової області (TDMA) із використанням маяків синхронізації та гарантованих часових інтервалів (GTS).

Існує чотири типи кадрів пакетів: дані, підтвердження, команда MAC і маяки. Кожен кадр містить послідовність синхронізації приймача, поле довжини пакета, адреси джерела та призначення, різні біти управління кадрами, корисне навантаження даних та послідовність перевірки кадрів (FCS), що виявляє помилки. Medium Access Control (MAC) генерує мережеві маяки над радіомеханізмами, якщо пристрій є координатором, він відповідає за синхронізацію з іншими маяками, підтримку зав'язків з Personal Area Network (PAN), управління доступом до каналу, обробку та підтримку механізму GTS і забезпечення надійного зв'язку між двома однорідними об'єктами MAC.

Серед технологій на базі стандарту IEEE 802.15.4, розглянуто наступні:

- LTE;
- Bluetooth;
- Ethernet;
- Wi-Fi.

Використання одразу декількох каналів передачі даних в декілька разів збільшить інформативність системи за рахунок підвищення відмовостійкості. Також у вищенаведеній схемі пропонується використання сонячної батареї, що робить систему досить економічно доцільною (особливо для нашого регіону).

Алгоритм роботи запропонованої системи можна описати наступним чином:

1. Перше, що робить система — збирає данні прилеглих ділянок доріг про те, що відбувається в реальному часі у оточуючому середовищі. Цю інформацію система отримує із інтеграції давачів Холла встановлених під смугами руху (при необхідності можна виконати інтеграцію із сторонніми системами по API).

2. Мікропроцесор обробляє зчитану інформацію, а потім за допомогою програмного забезпечення створює план оптимізації руху автотрафіку по вулицях міста.

3. Наступним кроком система надсилає команди контролеру для керування системами інформування водіїв / зміненню світлофорних тактів.

4. І нарешті - передає інформацію про зчитані данні по завантаженості ділянки дороги:

- до сусідніх мікропроцесорних систем, щоб вони могли включити цю інформацію у свої плани по оптимізації трафіку, що дозволяє виконувати як автономне керування трафіком на ділянках доріг, так і виконувати оптимізацію на рівні мережі міста, в залежності від умов дорожнього руху у реальному часі.

- до розумних транспортних засобів, які можуть використовувати цю інформацію ця самостійного коригування маршрутів.

На наступному рисунку зображена схема роботи системи (рис. 1.7):

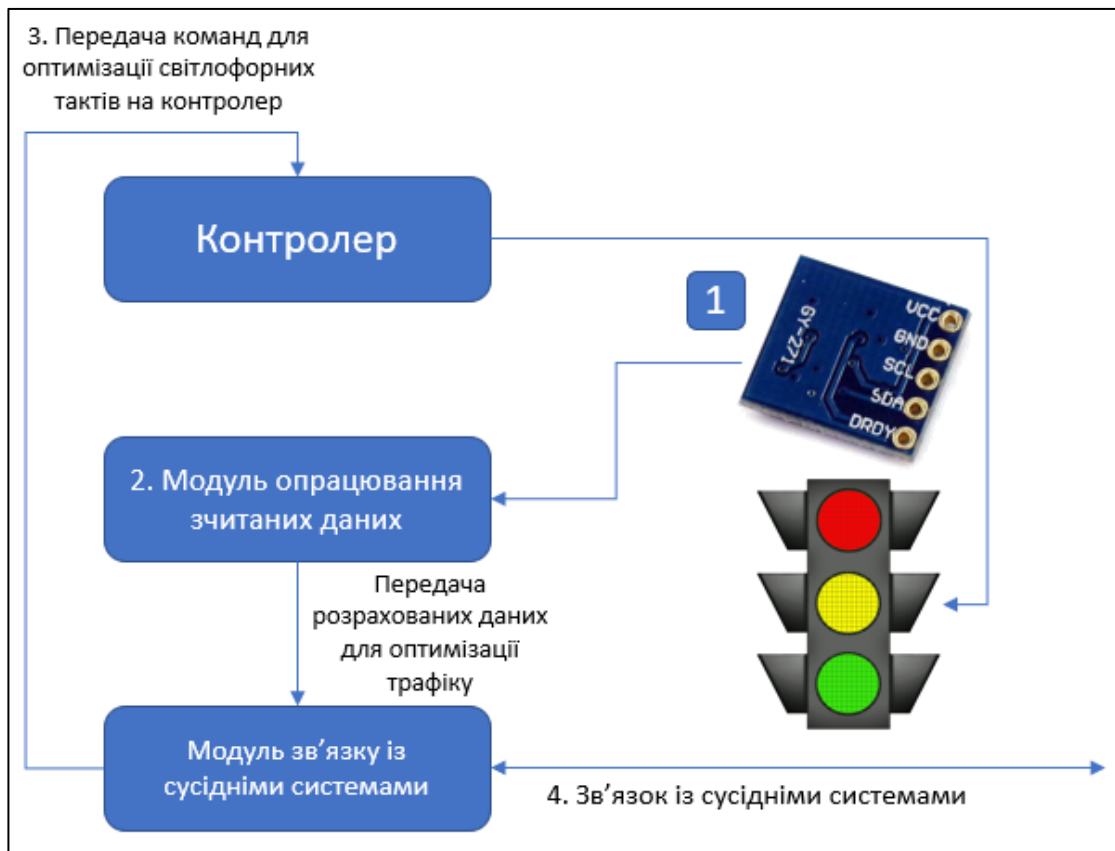


Рисунок 1.7. Система оптимізації трафіку на базі датчиків Холла.

1.5 Висновки до першого розділу

Злагожене функціонування транспортних магістралей можливе лише при оптимальному поєднанні способів моніторингу і оптимізації трафіку. Для цього необхідно ефективно оцінювати проблемні ділянки доріг, оперативно отримувати актуальну дорожньо-транспортну інформацію, і як наслідок ефективно керувати автомобільним рухом у містах (керування тривалістю світлофорних режимів та своєчасного вмикання інформаційних табло).

Сучасні системи моніторингу дорожнього руху мають ряд недоліків які впливають на якість фіксації інтенсивності трафіку і подальшої обробки даних. Зазвичай це проявляється при паркуванні автомобілів на смугах руху, чи при перекритті транспортним засобом сенсора / візуального спостерігача (зазвичай при багаторядному русі).

Аналіз існуючих систем для фіксації руху транспортних засобів показав, що вони мають ряд суттєвих недоліків. Вузьким місцем таких систем було визначено саме давачі які використовуються цими системами для фіксація руху автомобілів.

На базі виконаного пошуку систем оптимізації автотрафіку і патентного пошуку було прийняте рішення про використання давачів магнітного поля на базі ефекту Холла.

Ґрунтуючись на можливостях давачів Холла запропоновано систему визначення завантаженості і оптимізації руху автомагістралей, що має змогу фіксувати рух автомобілів при русі у декілька смуг, та русі транспортних засобів між смугами руху.

РОЗДІЛ 2. РОЗРОБКА ПРОЕКТУ ІoT-РІШЕННЯ ТА МЕТОДІВ ОБРОБКИ ДАНИХ В ІoT

2.1. Розробка проекту ІoT системи оптимізації трафіку

В ході виконання роботи був обраний метод фіксації руху транспортних засобів заснований на збуренні автомобілем магнітного поля Землі в процесі руху (кузов автівок містить феромагнітні елементи: двигун, передня і задня осі...). Основними проблемами які було виявлено у аналогів запропонованої в ході системи можна назвати — унеможливлення фіксації руху при перекритті візуальних давачів фіксації руху / при паркуванні автівок на проїзній частині / проблеми з фіксацією руху при перевлаштуванні транспортних засобів.

Метод фіксації збурення магнітного поля базується на чуттєвих датчиках магнітного поля — давачах Холла. Проект реалізовано на датчиках НМС5883L. Дана модель також називається «трьохосьовий компас», система заснована на основі НМС5883L здатна вимірювати напруженість магнітного поля уздовж всіх трьох осей. До основних характеристик давача можна віднести:

- межа вимірювання магнітного поля: ± 8 Гаусс;
- розрядність перетворення: 12 біт;
- точність: ± 2 мілігаусса;
- інтерфейс: I2C;
- напруга живлення: 3.3-5 вольт;
- підтримка автоматичного калібрування;
- розмір: 18x14x5 мм.

Саме можливість давачів Холла фіксувати збурення магнітного поля одразу по трьох координатах і можливість автоматичного калібрування відіграє ключову роль у функціонуванні системи. Для того, щоб впевнитись, що давачі дійсно фіксують збурення по трьом осям був проведений експеримент, результат якого наведено на наступному рисунку (рис. 2.1).



Рисунок 2.1. Фіксація збурень по трьом осям.

В ході експерименту над датчиком Холла був пронесений металічний об'єкт. На вище поданому рисунку можна побачити збурення магнітного поля по осям X, Y і Z. Судячи з цього можна зробити висновок, що припущення висунуте при виборі давачів було вірним в повній мірі.

Наступним кроком проектування стало створення структурної схеми системи (рис. 2.2) з урахуванням наявності наступних модулів:

- зчитування збурень магнітного поля;
- обробки (оптимізації) зчитаних даних;
- блок керування світлофорними режимами;
- блок обміну інформацією із навколишніми системами.

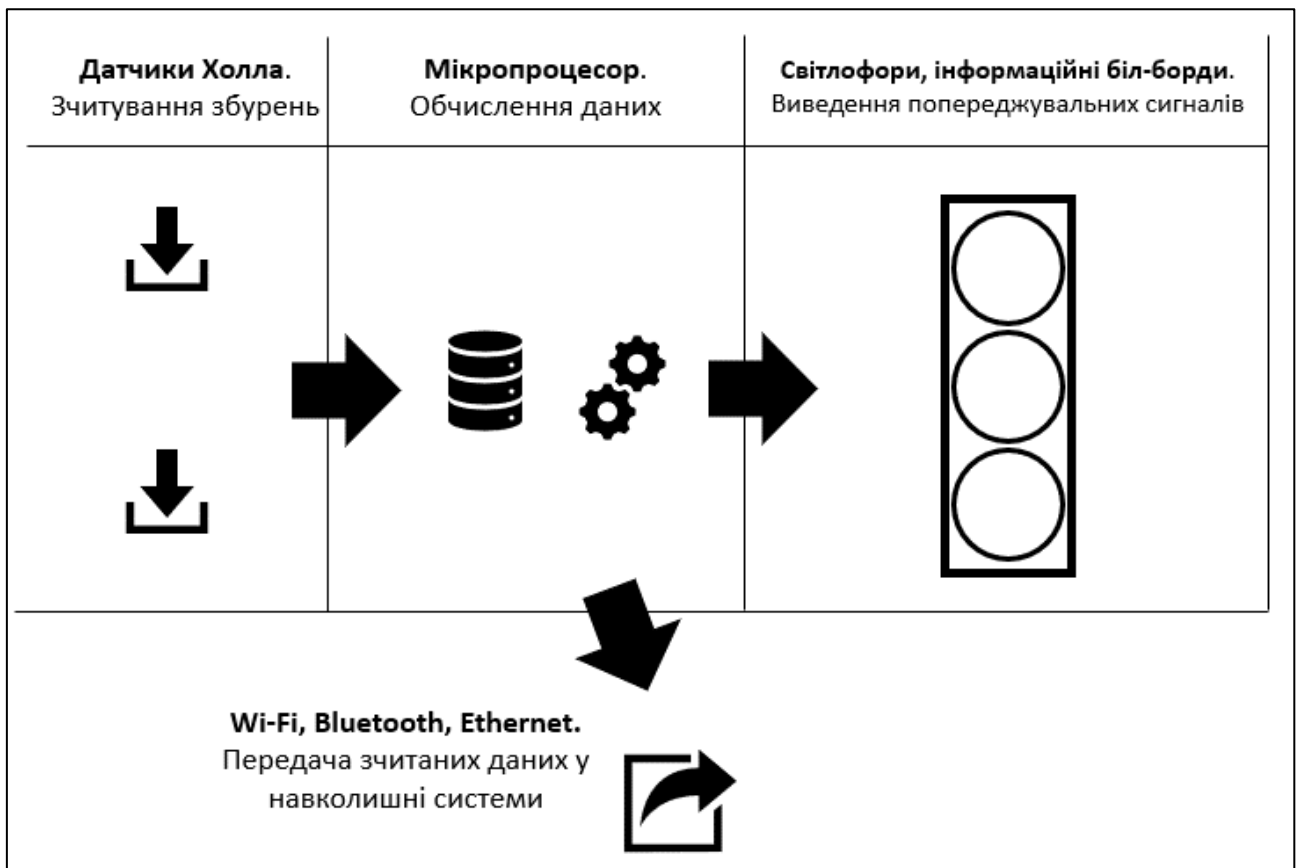


Рисунок 2.2. Структурна схема системи оптимізації трафіку.

Для перевірки описаного вище методу збору інформації про рух транспортних засобів, у другому розділі буде представлено лабораторний стенд зібраний для перевірки працездатності системи.

2.2. Методи та засоби обробки даних з давачів Холла

Датчики Холла складаються з декількох частин. По-перше, вони містять елемент Холла, який видає напругу Холла (V_H), що створюється за допомогою ефекту Холла [15]. По-друге, вони містять мікросхему Холла, яка перетворює вихідний сигнал Холла в цифровий сигнал високого / низького рівня в залежності від інтегральної схеми. По-третє, вони містять лінійну інтегральну схему Холла, яка підсилює і лінійлізує вихідний сигнал Холла.

Датчик видає вихідну напругу (рис. 2.3) пропорційна величині магнітного поля, вертикально прикладеної до датчика (може видавати позитивні і негативні

напруги відповідно до напрямку магнітного поля). Вихідна напруга при відсутності вертикального магнітного поля складає 0 В.

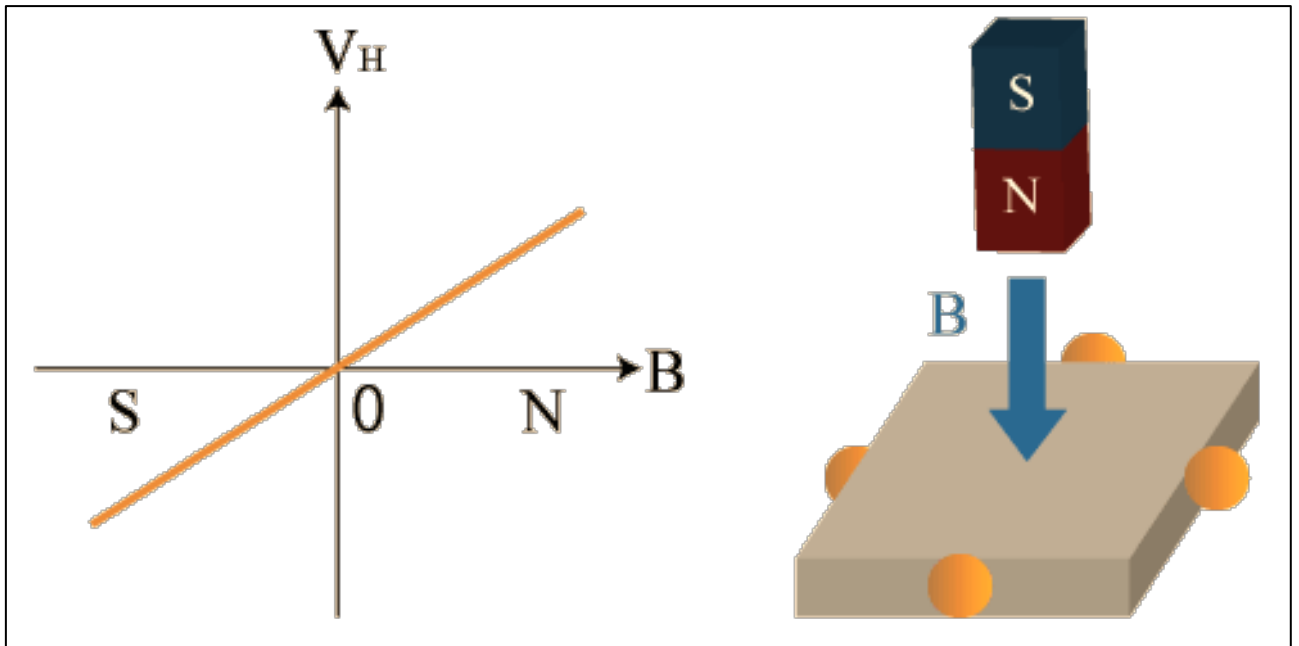


Рисунок 2.3. Вихідна характеристика елемента Холла.

Елемент Холла має всього чотири клеми: дві клеми GND для входу і дві диференціальні вихідні клеми (рис. 2.4).

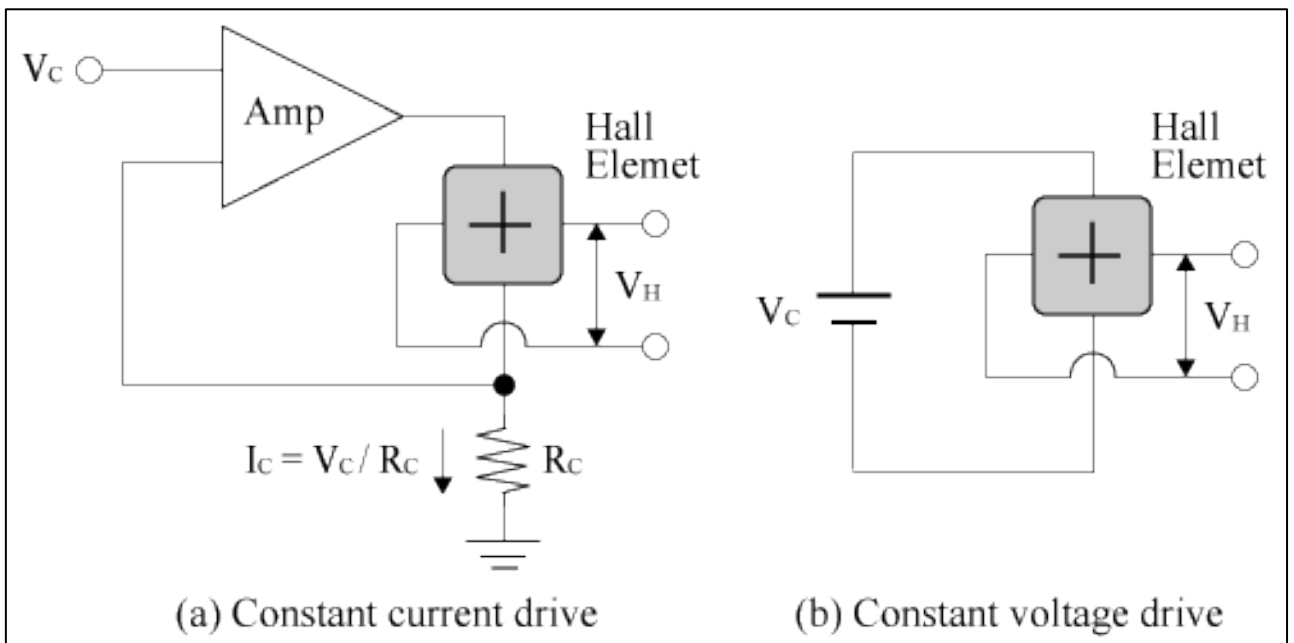


Рисунок 2.4. Принципова схема роботи датчика Холла.

Інтегральна схема Холла порівнює вихідний сигнал елемента Холла з певним граничним значенням і виводить його як високий або низький сигнал.

Оскільки діапазон вихідної напруги регулюється блоком живлення, до цього виходу можна легко підключити мікропроцесор.

Вихідна напруга визначається як висока або низька відповідно до величини магнітного поля, вертикально прикладеного до датчика. Є три типи результатів визначення полюса:

- S-полюс;
- N-полюс;
- біполярність.

Коли величина магнітного поля перевищує V_{OP} , вихідна напруга стає низькою, а коли значення магнітного поля стає нижче V_{RP} , вихідна напруга стає високою (рис. 2.5). У цьому випадку між V_{RP} і V_{OP} виконується умова « $V_{RP} < V_{OP}$ (з гістерезисом)».

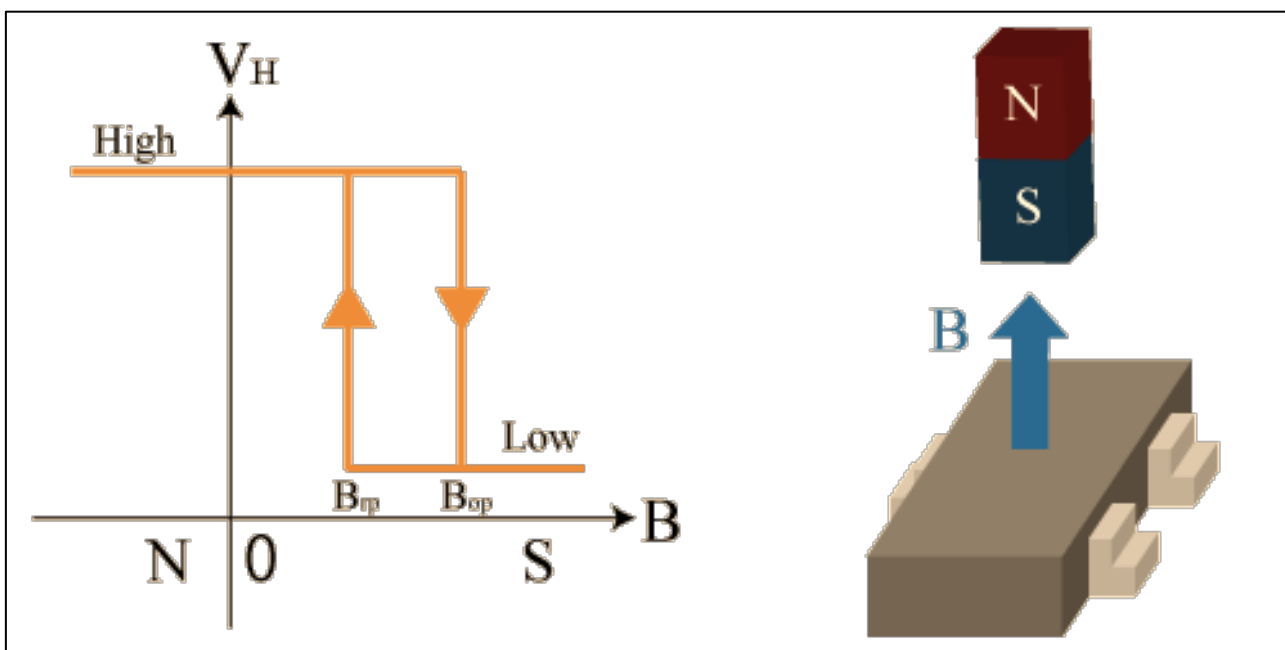


Рисунок 2.5. Вихідні характеристики інтегральної схеми Холла при виявленні південного полюса.

Лінійна інтегральна схема Холла має дві вхідні клеми, VCC і GND, і одну вихідну клему. Підключення інтегральної схеми (IC), як показано на наступному рисунку (рис. 2.6), до елемента Холла реалізує інтегральний струм Холла. Схема працює від постійної напруги.

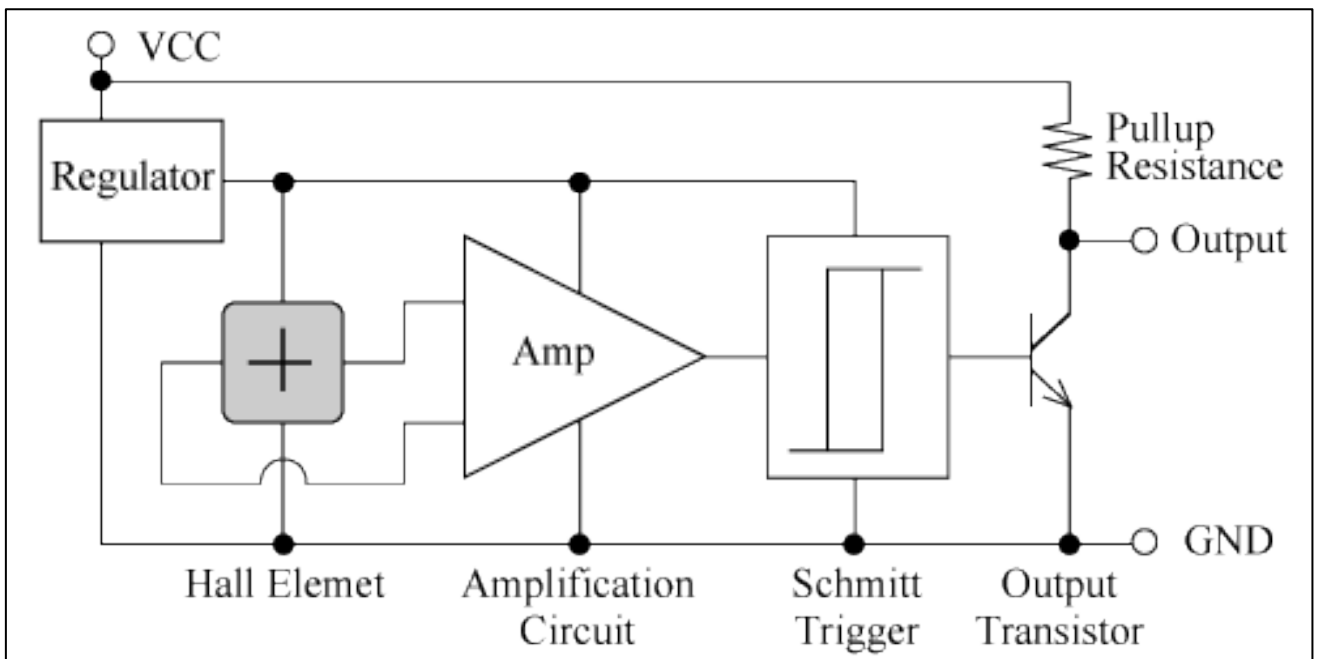


Рисунок 2.6. Принципова схеми роботи датчика Холла із підключенням ІС.

Лінійний інтегральний струм Холла застосовує коефіцієнт посилення до виходу елемента Холла, що призводить до лінійзації вихідних значень. Оскільки діапазон вихідної напруги регулюється джерелом живлення, до наступного каскаду можна підключити мікроконтролер [16].

2.3. Обґрунтування обраних технологій для створення інформаційної системи

2.3.1 Обґрунтування вибору Arduino Pro Mini

Для реалізації проекту була обрана відлагоджувальна плата Arduino Pro Mini у зв'язку із своїми малими габаритами, великою кількістю цифрових і аналогових виходів, можливістю обміну даними по інтерфейсу I2C і функціоналом роботи з широтно-імпульсною модуляцією сигналів [17].

Arduino Pro Mini підтримує ряд можливостей для здійснення комунікації з ПК, іншим Arduino або мікроконтролером. ATmega328 оснащена приймачем UART, що може виконувати послідовну передачу даних за допомогою цифрових виводів RX і TX. У базове програмне забезпечення для роботи з Arduino входить Arduino IDE, що дозволяє писати код на мові C або C++ і завантажувати його на

контролер, читати і передавати прості текстові дані на плату Arduino через USB-місток. Сама платформа Arduino гарно документована і має велику кількість гарно описаних і що важливіше – вільно доступних бібліотек. Бібліотека SoftwareSerial надає можливість здійснювати послідовний зв'язок на цифрових виводах Arduino Pro Mini [18].

У мікроконтролері ATmega328, яким оснащена Arduino Pro Mini, присутнє підтримка інтерфейсів I2C і SPI. Бібліотека Wire спрощує роботу з інтерфейсом I2C.

Мікроконтролер ATmega328, який використовується в Pro Mini надається з передвстановленим bootloader-ом, що спрощує завантаження в мікроконтролер нового коду. Взаємодія з bootloader-ом здійснюється по протоколу — STK500.

2.3.2 Обґрунтування вибору середовища для візуалізації експериментальних даних — LabVIEW

Для візуалізації отриманих в ході експериментальної перевірки результатів вимірювання збурень магнітного поля було вирішено обрати графічне середовище для розробки — LabVIEW.

LabVIEW створювалася для інженерів, а не для програмістів. Тому це середовище (і мова програмування G) зроблені максимально інтуїтивно зрозумілими. Тому швидкість його освоєння просто приголомшлива. Враховуючи те, що LabVIEW – це графічна мова програмування, немає необхідності витратити час на перевірку синтаксису, пошук помилок і друкарських помилок. LabVIEW оперує "об'єктами".

LabVIEW було створено для написання програма для автоматизації в промисловості і навчальних лабораторій, тому для роботи з периферією додаток в своєму складі має дуже великий набір бібліотек (робота з RS-232 — COM порт, LPT, USB, протоколи TCP / IP, UDP і т.д.).

Зараз дуже багато відомих виробників плат збору даних, камер машинного зору, промислового обладнання поставляються разом з набором готових бібліотек для їх обладнання в LabVIEW. Що дозволяє в стислі терміни

розпочати роботу з обладнання і налаштувати його під свої завдання.

Дане середовище зручно використовувати для швидкого і якісного опрацювання зчитаних даних з давачів. LabVIEW надає можливість будувати широкий сект вимірювальної техніки з використанням об'єктно-орієнтовного способу програмування. Більш того, платформа окрім зчитування даних і виконання проміжних розрахунків, дозволяє одночасно будувати візуалізацію розрахованих результатів!

Для обраної системи візуалізації отриманих даних було висунуто чіткі вимоги, а саме реалізувати:

- реалізувати можливість мануального керування вимірюванням швидкості (вмикання/вимикання);
- реалізувати запис зчитаних даних у файл;
- логування можливих помилок;
- реалізації кнопки «STOP» для зупинки програми;
- побудова графіку виміру збурень магнітного поля;
- реалізація індикації напрямку руху авто;
- відображення швидкості руху транспортного засобу.

Поставлені задачі було в повній мірі реалізовано, що буде відображено в розділі 4.

2.4. Обґрунтування обраного методу оптимізації синхронізації дорожнього сигналу

Сучасні технології управління сигналами дорожнього руху мають низьку ефективність, у пікові проміжки навантаження на автомагістралі. На даний момент було запропоновано велику кількість методів для підвищення ефективності систем керування дорожніми сигналами в умовах високої завантаженості доріг [19-21]. Однак дуже мало з цих методів можуть бути широко використані через специфічні вимоги або обмеження характерні для цих методів [22]. Для того, щоб встановити один конкретний метод управління

дорожнім рухом, що буде здатний ефективно працювати в моменти «пікових» навантажень, слід виявити природу управління дорожнім сигналом.

Методи керування сигналами дорожнього руху намагаються встановити зв'язок між спостережуваними параметрами трафіку, такими як кількість авто, затримка в русі та довжина черги з авто, та такими параметрами систем сигналів дорожнього руху, як послідовність фаз і тривалість циклу. Таким чином, різні алгоритми оптимізації сигналів дорожнього руху шукають «баланс» між значеннями параметрів сигналів дорожнього руху, щоб отримати оптимальні значення одного або декількох параметрів дорожнього руху, розглядаючи параметри дорожнього сигналу як незалежні змінні в конкретних умовах дорожнього руху. Цього принципу дотримуються майже всі загальнозживані методи оптимізації управління дорожнім рухом, які включають метод TRRL (Transport and Road Research Laboratory) [23], метод HCM (Highway Capacity Manual) [24] та адаптивне програмне забезпечення для управління дорожнім рухом, таке як SCATS (Sydney Coordinated Adaptive Traffic System) [25] та SCOOT (Split Cycle Offset Optimizing Technique) [26].

Можна побачити, що описи характеристик потоку руху є найважливішим фактором оптимізації синхронізації дорожнього сигналу. Однак транспортний потік стає нестабільним, коли «попит» транспорту наближається або перевищує пропускну здатність [27]. Шкідливі наслідки, такі як «залишкова черга» або «постійний червоний», ускладнюють точну характеристику потоків руху. Таким чином, методи оптимізації сигналів дорожнього руху, встановлені формулами потоку руху в нормальних умовах руху, не підходять для перенасичених умов. Евристичні алгоритми, керовані даними, можуть бути ідеальним методом для отримання оптимізованого плану синхронізації дорожнього сигналу в перенасичених умовах. Евристичні алгоритми не покладаються на формули потоку руху, а шукають сценарії оптимізації на основі даних про трафік у реальному часі.

Основною метою даного розділу є представлення багатоцільового методу оптимізації для отримання відносно кращого алгоритму синхронізації сигналу для перенасиченого перехрестя. Перш ніж розробляти алгоритм, аналізуються характеристики перенасиченого потоку транспорту для отримання оптимізованих цілей алгоритму. Потім ми представляємо деталі алгоритму, які включають схему кодування, цілі оптимізації та вибір алгоритму. Нарешті результати збіжності та моделювання алгоритму за різних умов узагальнюються та аналізуються.

Статус руху може бути визначений за інтенсивністю руху, тобто співвідношенням V/C (обсягу трафіку до ємності дороги). На сигналізованому перетині перенасичення можна визначити як умову зупинки із «залишковою чергою» [28], що проілюстровано на наступному рисунку (рис. 2.7). Оскільки безпосередньо виміряти залишкову чергу важко, її можна оцінити за допомогою давачів Холла.

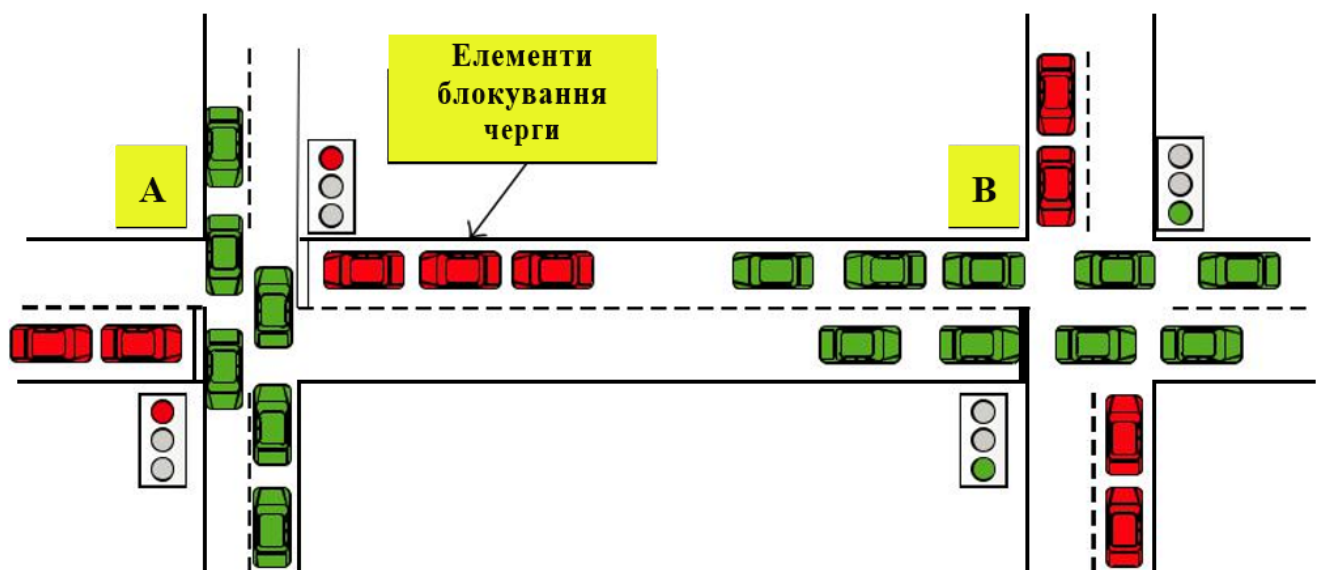


Рисунок 2.7. Залишкові черги на перетині.

Характеристики потоку руху будуть різними, коли транспортний потік перебуває в умовах перенасичення або наближення до умов насичення [29]. Коли стан дорожнього руху наближається до насичення, транспортний потік

стане нестабільним. Невеликі коливання будь-якого транспортного засобу взводу можуть спричинити несприятливі наслідки та різко знизити ефективність дорожньої системи [30]. Коли стан трафіку перебуває в перенасиченому стані, для оцінки довжини черги може бути прийнята модель детермінованої черги. З причини низької стабільності насиченого потоку транспорту параметри, оцінені моделями, перерахованими вище, навряд чи можуть бути використані в процесі синхронізації сигналу.

Як правило, перенасичений стан спочатку з'являється на одному або декількох перехрестях. Потім затори починають поширюватися на сусідні перехрестя із додатковим «шкідливим ефектом», таким як «заручники черги», що показано на наступному рисунку (рис. 2.8). Перехрестя вздовж маршруту з більшим обсягом руху можуть поширювати затори набагато швидше, ніж інші маршрути [31]. Нарешті, вся дорожня мережа буде перевантажена, і навіть перейде у стан «затор» [32]. Згідно з процесом утворення заторів, стратегії управління дорожнім рухом в перенасичених умовах слід застосовувати на ізольованих перехрестях, щоб уникнути шкідливих наслідків.

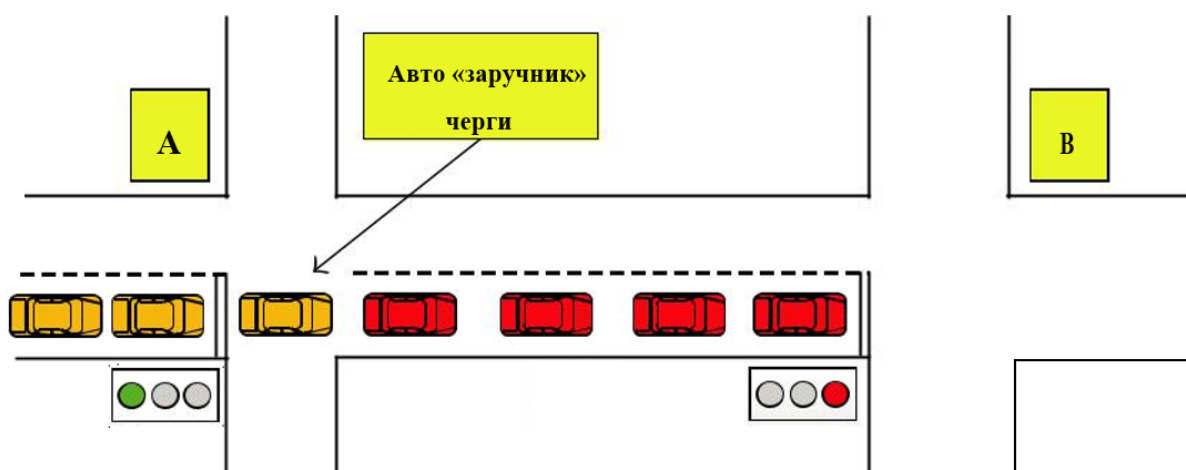


Рисунок 2.8. Затор на перетині.

2.4.1 Вибір та обґрунтування алгоритму оптимізації

Максимальна пропускна здатність і вимір співвідношення черг - дві суперечливі цілі, що є типовою багатоцільовою проблемою оптимізації (БПО) [33]. Багато інтелектуальних алгоритмів, таких як Evolutionary Algorithm (EA) [34], Particle Swarm Optimization (PSO) [35], Simulated Annealing Algorithm (SAA) [36] та алгоритм Ant Colony Optimization (ACO) [37], може бути використаний для отримання фронту Парето для БПО. Однак із усіх алгоритмів лише Genetic Algorithms (GA), як категорія еволюційного алгоритму, були успішно застосовані в системі управління дорожнім сигналом для комерційних виробництв [38]. Недомінований сортувальний генетичний алгоритм II (NSGA-II) [39] має кращу продуктивність, ніж інші багатоцільові еволюційні алгоритми [40], і, таким чином, він обраний як метод оптимізації управління дорожнім сигналом у цьому розділі. Алгоритм NSGA-II вдосконалив NSGA, додавши елітарну стратегію, стратегію оцінки щільності та швидку стратегію сортування без домінування.

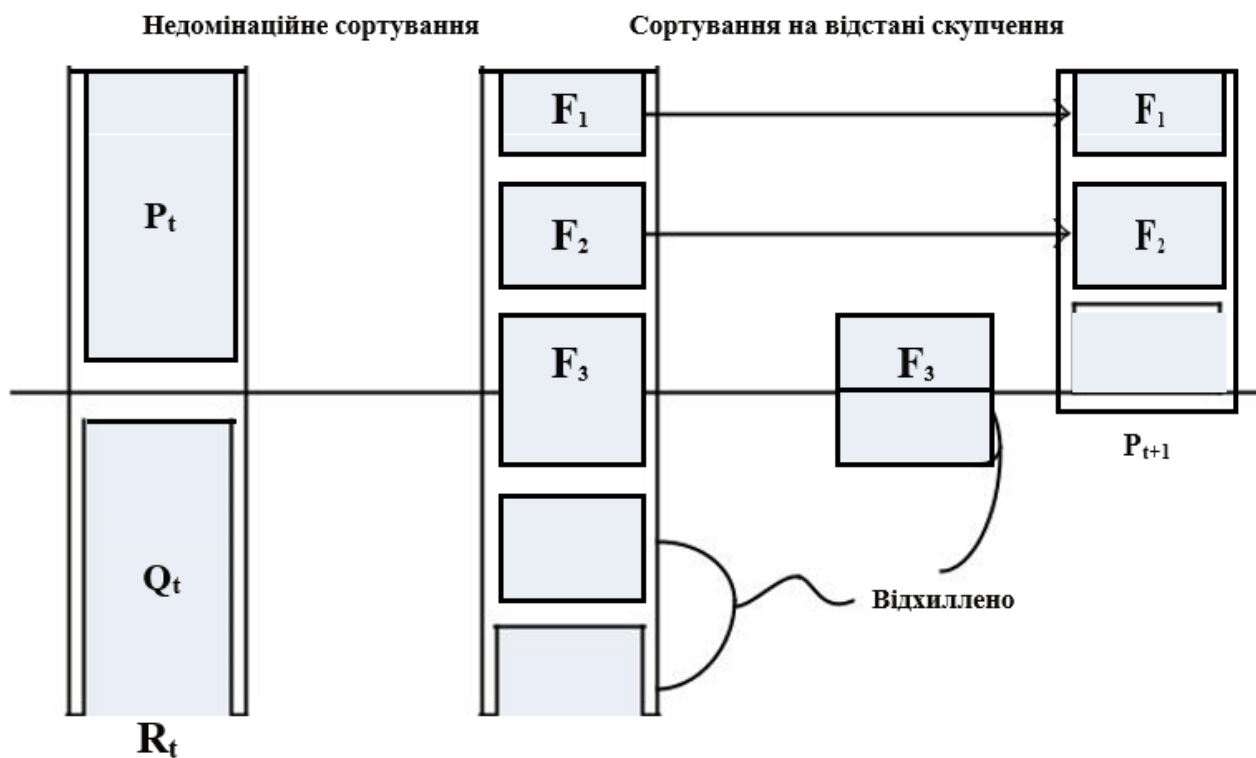


Рисунок 2.9. Основна процедура алгоритму NSGA-II.

Недомінаційне сортування та елітарна стратегія можуть покращити ефективність та допомогти нам отримати краще рішення. Стратегія оцінки щільності дозволяє уникнути параметра частки, який важко визначити. Ці стратегії роблять алгоритм NSGA-II кращим, ніж інші MOEA. Однак збіжність і різноманітність NSGA-II можуть бути гіршими через стратегію оцінки щільності. Основна процедура алгоритму NSGA-II показана на попередньому рисунку (рис. 2.9).

Основний цикл алгоритму NSGA-II перерахований нижче:

1. Спочатку створюється випадкова батьківська сукупність P_0 . Загальна кількість входжень сортується на основі недомінації. Кожному рішенню присвоюється придатність (або ранг), що дорівнює рівню його недомінації. Для створення популяції Q_0 потомства використовуються звичайні оператори відбору, рекомбінації та мутації в бінарних турнірах розміром N . Нехай $t = 0$.

2. Формується комбінована сукупність $R_t = P_t \cup Q_t$. Популяція R_t має розмір $2N$. Потім популяція R_t сортується згідно з недомінацією. Формується найкращий набір без домінування Fi .

3. Оператор переповненого порівняння $<_n$ вибирається для сортування бездомінантного набору Fi за спаданням. Найкращі популяції члени набору вибираються для нової сукупності P_{t+1} .

4. Нова сукупність P_{t+1} тепер використовується для відбору, кросоверу та мутації для створення нової популяції Q_{t+1} .

5. Коли досягається умова завершення, цикл зупиняється; інакше збільшуємо значення t на 1 і переходимо до кроку 2.

Схема кодування, очевидно, впливає на генетичні маніпуляції, особливо для кросоверу. Двійкове кодування, кодування реального числа та структурне кодування - це три основні базиси схем кодування в генетичних алгоритмах. У задачі оптимізації управління сигналами дорожнього руху схема кодування повинна отримати можливість описувати план синхронізації

сигналу. Якщо застосовується двійкове кодування, необхідно накласти додаткові обмеження в процесі кросинговеру та мутації. Крім того, слід також враховувати логічні обмеження між генами (тривалість циклу має бути підсумовуванням усіх «зелених часів» та «часів простою»). Всі ці обмеження збільшують складність розрахунку. Отже, кодування реального числа обрано як схему кодування в задачі оптимізації управління сигналами руху. Зелений час кожної фази є виконуваним генетичним оператором гена.

Враховуючи, що стратегії управління дорожнім рухом для перенасичених умов м'яко кажучи «неготові», краще зберегти поточну стратегію контролю, а не застосовувати нову. Важливо також застосовувати стратегії (де це можливо), щоб запобігти перенасиченню, а не реагувати на проблеми після факту. Як тільки попит на дорожній рух повернеться до нормального рівня, стратегія управління дорожнім рухом у «звичайному» режимі знову досить легко зможе керувати потоком руху. Таким чином, максимальну пропускну здатність слід обрати однією з цілей оптимізації.

Проблема черги - це основна проблема в умовах перенасиченого трафіку. Алгоритм оптимізації синхронізації сигналів повинен мати можливість підтримувати співвідношення черг кожного напрямку в прийнятному діапазоні. У цій стратегії управління «зелений час» слід скоригувати, щоб збалансувати черги на підходах до перехрестя під час перенасичених умов. Завдання - мінімізувати кількість заблокованих перехресть за допомогою оптимізації черг критичного перехрестя.

На основі перерахованого вище, в запропонованому алгоритмі оптимізації синхронізації трафіку обрано дві цілі оптимізації, «максимум пропускну здатності» та «оптимізація показника черг».

Для того, щоб досягти завдань оптимізації алгоритм прогнитує збільшити тривалість циклу, щоб уникнути втраченого часу, спричиненого процесом фазового переходу. Однак черга на підходах до перехрестя сформується дуже швидко в умовах перенасиченості та отримає надзвичайно

високу ймовірність швидкого розповсюдження, якщо обрано довгу тривалість циклу. На даний момент запропоновані алгоритми повинні мати можливість підтримувати коефіцієнт черги при кожному підході для запобігання шкідливих наслідків. Передбачається, що дані про трафік з високою роздільною здатністю доступні для оптимізації плану синхронізації сигналів. Результатом роботи алгоритму будуть «зелені часи» для кожної. Послідовність фаз визначено заздалегідь.

У перенасичених умовах система дорожнього руху нестабільна, особливо на критичному шляху підмережі або критичному русі по перехрестю. Отже, основною метою контролю дорожнього руху в перенасичених умовах є підтримка транспортного потоку в стабільному стані та пропускання якомога більшої кількості транспортних засобів, особливо для критичних ділянок. З метою максимізації пропускної здатності критичного руху ваговий коефіцієнт слід помножити на фактичні числа пропускної здатності у відповідній цільовій функції оптимізації. Таким чином, функцію оптимізації пропускної здатності в алгоритмі можна отримати за формулою 2.1:

$$O_c = \max \sum_{i=1} (\omega_t * n_t(g_i, C, t)), \quad (2.1)$$

де O_c — зважена максимально допустима кількість транспортних засобів на перехресті;

ω_t — ваговий коефіцієнт кожного моменту часу;

$n_t(g_i, C, t)$ — фактична пропускна спроможність транспортних засобів.

Багато заходів щодо оптимізації трафіку можна класифікувати як стратегію обслуговування черги, наприклад середню довжину черги, максимальну довжину черги, середню максимальну довжину черги та розсіяну довжину черги. Коефіцієнт заповнення черги використовується як ціль оптимізації з урахуванням впливу прийнятного простору черги. Впроваджуючи цю концепцію в цільову функцію оптимізації, можна ефективно пом'якшити збитковий ефект. Тим часом довжина черги кожної фази також може бути

оптимізована. Функцію оптимізації обслуговування коефіцієнта черги в алгоритмі можна отримати за формулою 2.2:

$$Q_t = \min \frac{1}{P} \sum_{p=1}^P \frac{len_p}{L_p}, \quad (2.2)$$

де P — кількість фаз;

$len_p(t)$ — довжина черги на фазі « P »;

L_p — максимально допустима довжина черги відповідної фази.

Проаналізувавши збіжність та результати запропонованого алгоритму при різних номерах генів та обсягах трафіку пропонується використовувати цей алгоритм як еталонний.

2.5. Висновки до другого розділу

В ході виконання роботи був обраний метод фіксації руху транспортних засобів заснований на збуренні автомобілем магнітного поля. Метод фіксації збурення магнітного поля базується на чутливих датчиках магнітного поля — давачах Холла. Проект реалізовано на датчиках HMC5883L.

В ході виконання даного розділу обґрунтовано і доведено необхідність використання давачів Холла у зв'язці із мікропроцесорним контролером Arduino Pro Mini.

Для візуалізації результатів вимірів збурень магнітного поля було вирішено використовувати середовище об'єктного-орієнтованого програмування — LabVIEW. Дане середовище дозволяє в короткі строки розробити інтерфейс який буде включати до свого складу усі необхідні «лабораторні стенди» для відображення всіх параметрів які зчитуються давачами і мають бути розраховані по певним алгоритмам. Так як LabVIEW - графічна мова програмування, тому не потрібно витрачати час на перевірку синтаксису, пошук помилок і помилок написання коду. Завдяки великій кількості загальнодоступних бібліотек вдалося виконати підключення усіх сторонніх систем.

В якості методу оптимізації синхронізації дорожнього сигналу було обрано і детально розглянуто алгоритм NSGA-II.

Для реалізації кінцевої системи оптимізації трафіку необхідно:

- створити програмне забезпечення для мікроконтролера на платформі Arduino IDE;
- реалізувати зчитування даних про збурення магнітного поля по трьох координатах кожного з датчиків;
- реалізувати алгоритм оптимізації трафіку;
- візуалізувати отримані дані з супутнім розрахунком швидкості і напрямку руху транспортного засобу.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ АРХІТЕКТУРИ ІОТ-РІШЕННЯ

3.1. Архітектура мікропроцесорної системи

Проект по оптимізації дорожнього трафіку мегаполісів вирішено будувати на базі:

- двох датчиків Холла;
- мікропроцесорного мікроконтролера Arduino Pro Mini;
- LTE / Bluetooth модуля;
- джерела живлення.

Для того, щоб завантажити програмний код в мікропроцесор Atmega328 встановлений на мікроконтролері Arduino Pro Mini необхідно виконати підключення через USB-місток (рис. 3.1).

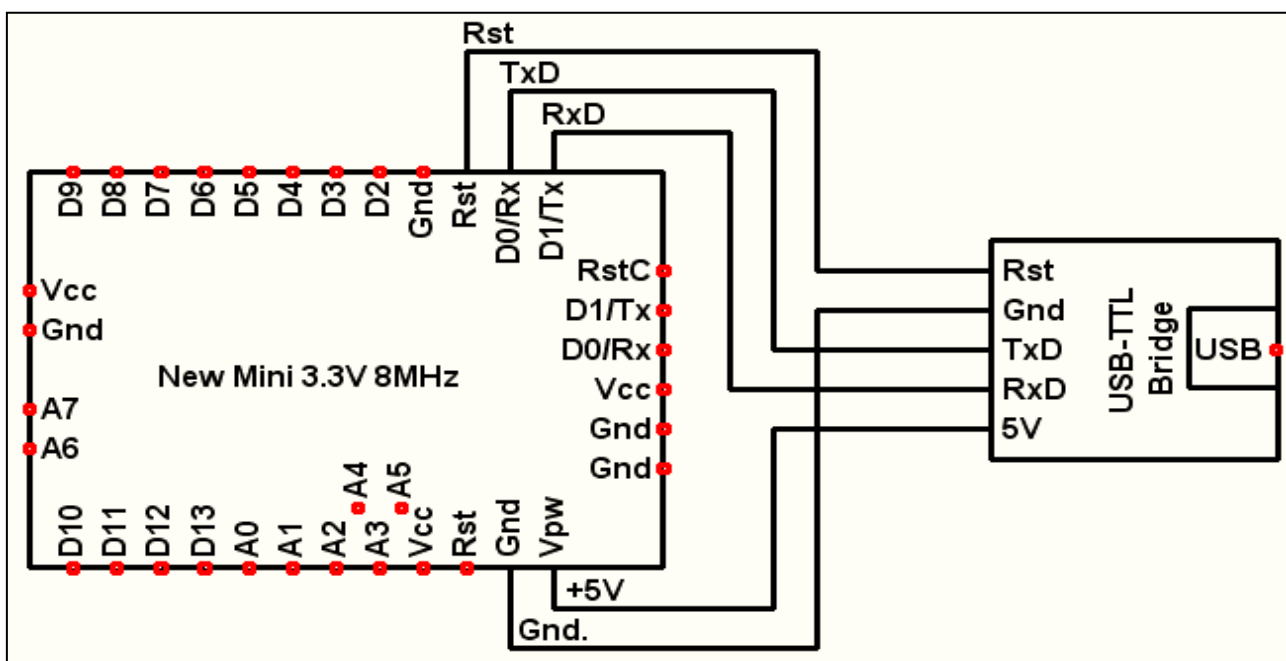


Рисунок 3.1. Схема підключення плати Arduino Pro Mini до USB-містка.

На схемі показано підключення Arduino до переробленої плати-адаптера USB-TTL-SJ. Переробка полягає в заміні лінії малопотужного живлення 3.3V на лінію R_{ST}-скидання, що дозволяє автоматично завантажувати програми в процесор без натискання кнопки на платі Arduino. Таблиця співставлення

виводів USB-містка і Arduino (Таблиця 3.1).

Таблиця 3.1 — Співставлення виводів

	Вивід 1	Вивід 2	Вивід 3	Вивід 4	Вивід 5
USB	+5V	Gnd	RxD	TxD	Rst
Arduino	Vpw	Gnd	Tx	Rx	R _{ST}

- TxD — передача сигналу від комп'ютера на прийом мікроконтролером Rx;
- Tx — передача сигналу від мікроконтролера на прийом комп'ютером RxD;
- R_{ST} — скидання процесора.

Для плати Arduino — значення напруги (3.3V) на виводах Vcc визначається стабілізатором на платі. Схема розміщення виводів плати (рис. 3.2) Arduino Pro Mini.

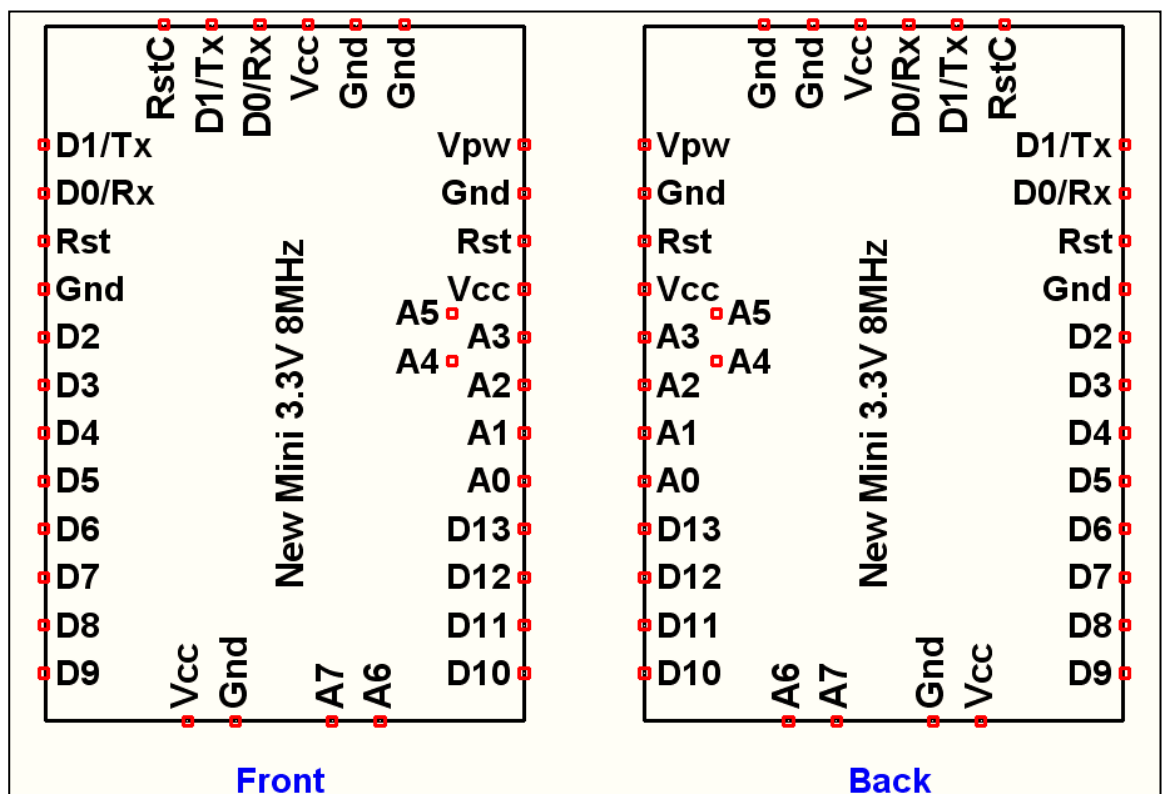


Рисунок 3.2. Вид на виводи Arduino згори і знизу, відповідно.

Дріт R_{ST} - скидання підключається до Arduino тільки для переробленого USB-адаптера. Кабель надівається на плату з боку протилежного до кнопки так, що перший ряд виводів залишився вільним, кабель має бути орієнтований назовні від плати (рис. 3.3).

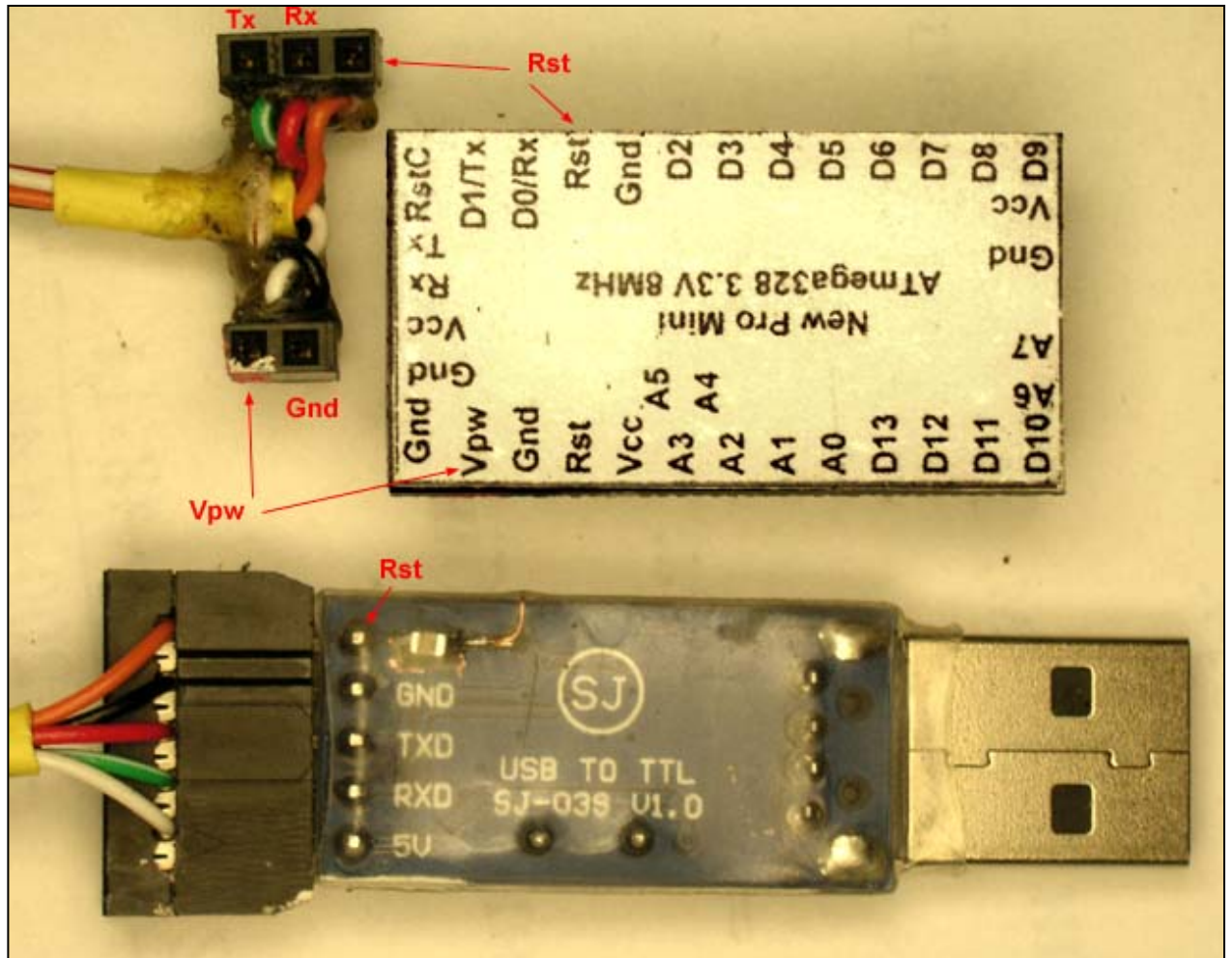


Рисунок 3.3. Плата Arduino і USB-адаптер з шнуром, вид знизу.

Після реалізації апаратної частини для прошивки мікроконтролеру була реалізована схема підключення давачів Холла (рис. 3.4), сама схема представлена на наступному рисунку.

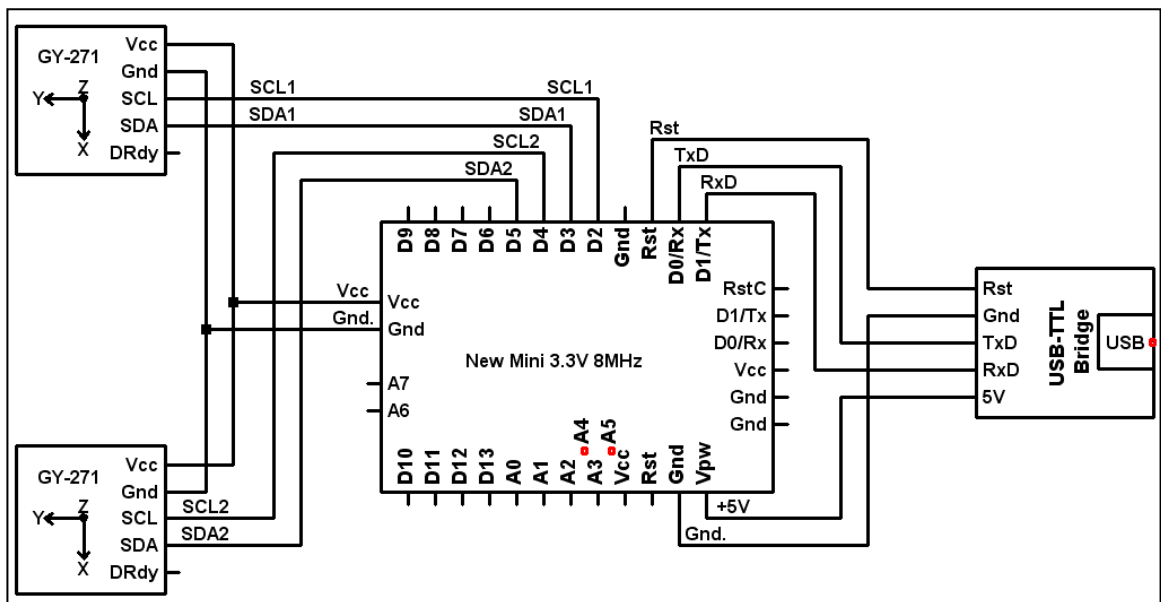


Рисунок 3.4. Схема електрична принципова підключення давачів Холла.

Для передачі даних для подальшої обробки на ПК була розроблена програма для мікропроцесорної плати. Детальний текст програми для Arduino написаний мовою C наведено у «Додатку А».

Програма за допомогою програмної оболонки Arduino IDE завантажується в пам'ять мікроконтролера. На наступному рисунку наведено приклад вікна середовища для програмування Arduino з програмним кодом для передачі результатів реєстрації магнітних збурень на ПК (рис. 3.5).

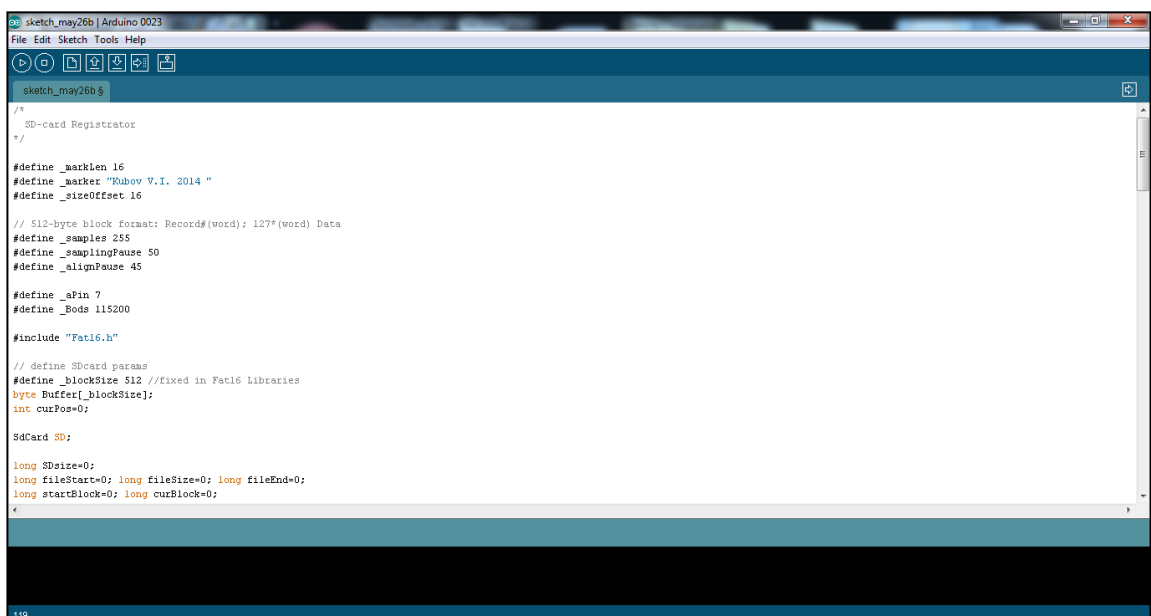


Рисунок 3.5. Середовище Arduino IDE.

Дана програма призначена для вимірювання збудження магнітного поля Землі викликаною переміщенням феромагнітного об'єкта над давачами чутливими до зміни магнітного поля. В якості датчиків фіксації зміни магнітного поля використовується пара цифрових трьох-координатних давачів НМС5883L.

Оболонка мікропроцесора опрацьовує зчитані данні, на їх основі вираховує швидкість руху об'єкта, його напрям руху і феро-магнітні нариси. Наступним кроком система оптимізує тривалість світлофорних тактів за рахунок розрахунку завантаженості трафіку на прилеглих ділянках доріг. На останок контролер передає розраховані данні по LTE або Ethernet каналу у глобальний центр обробки, який в свою чергу виконує оптимізацію трафіку міста по математичній неоднорідній моделі Пуасона.

3.2. Комунікаційні технології та системи використані у проєкті

Зазвичай у інтернеті речей зв'язок відбувається через пристрої або шлюзи. Найбільш уживаним рішенням є шлюзовий зв'язок. Тобто може відбуватися паралельне «спілкування» через декілька інтерфейсів. Існує великий вибір технологій бездротового зв'язку. Важливим параметром для вибору технології зв'язку є пропускна здатність, показники споживання енергії та безпека передачі даних.

У сучасних бездротових технологіях в основному використовуються такі способи передачі даних [41]:

- GSM (телеметрія, smart city, пульти дистанційного керування, збір середніх об'ємів даних);
- LTE (телеметрія, smart city, пульти дистанційного керування, збір великих об'ємів даних);
- Bluetooth (домашня автоматизація - smart home);
- WiFi (домашня автоматизація - smart home);
- Lora / LoRaWAN (телеметрія, лічильники енергії, передача даних датчиків, smart city);

- ZigBee (телеметрія, лічильники енергії, передача даних з датчиків);
- 6LoWPAN (телеметрія, smart home, пульти дистанційного керування, збір малих об'ємів даних);
- Thread (телеметрія, пульти дистанційного керування, збір малих об'ємів даних).

Розглянемо детальніше технологію Bluetooth. Вона призначена для бездротового зв'язку безлічі електронних пристроїв короткого діапазону. Починаючи з 1994 року, ця технологія розвивалася і в даний час використовує версію 4.2. Вона працює в смузі частот від 2402(ГГц) до 2483(ГГц). Ця версія забезпечує доступ розумних пристроїв до Інтернету без необхідності додаткового, проміжного обладнання. Найвища швидкість передачі - 24 Мбіт/с. На початку 2017 року була запущена версія Bluetooth 5, яка заявила в кілька разів вищу швидкість передачі та дальність передачі.

До переваг технології Bluetooth можна віднести:

- помірне споживання електроенергії у порівнянні з Wi-Fi;
- можливість безпосереднього спілкування зі смартфонами та іншою «розумною» електронікою.

Недоліки технології Bluetooth:

- короткий діапазон роботи;
- платна ліцензія;
- використання досить перевантаженого діапазону частот 2,4(ГГц).

Другою технологією яку було обрано для використання у проєкті стала — 4G (LTE). Це технологія четвертого покоління мобільних технологій, яку також називають LTE (Long Term Evolution). Вона призначена для високошвидкісного інтернету в мобільних мережах. Це до 10 разів швидше, ніж мережі 2G і 3G. Смуга частот для Європи 700-2600 (МГц) зі швидкістю передачі 70-250 (Мбіт/с).

Переваги технології 4G:

- висока швидкість передачі;
- швидка побудова і розгортання мережі;
- перспективний частотний діапазон.

Недоліки технології LTE:

- високе енергоспоживання;
- досить висока ціна модулів.

Технології Wi-Fi та Bluetooth (краще підходять для локальної мережі) - це протоколи, які в основному використовуються для спілкування з персональними пристроями. 2G / 3G / 4G (стільникові мережі) - це мобільні технології, які підходять для великої пропускної здатності великої кількості пристроїв [42].

3.3. Проміжне програмне забезпечення

Проміжне програмне забезпечення для обміну повідомленнями - це програмний рівень або сукупність підрівнів, що з'єднує різні різноманітні домени. Проміжне програмне забезпечення для обміну повідомленнями - це «клей», який утримує все це разом. Дослідження систем проміжного програмного забезпечення з роками набирають обертів. Однією з важливих переваг системи проміжного програмного забезпечення є її здатність забезпечувати безперебійну взаємодію між різними компонентами. Це дозволяє програмісту зосередитись на створенні стандартних, адаптованих та ефективних рішень, а не турбуватися про конкретні деталі основних рівнів. Існують різні стандарти та протоколи для побудови систем проміжного програмного забезпечення, орієнтованих на передачу повідомлень.

Існують різні стандарти та протоколи для побудови систем проміжного програмного забезпечення, орієнтованих на повідомлення. Однією з найпопулярніших проміжних програм є служба обміну повідомленнями Java. JMS забезпечує стандартну API інтеграцію для платформи Java, а також багато

додаткових служб для взаємодії між платформами Java та поза ними. Інтеграція з іншими мовами програмування, такими як Ruby, Scala, тощо, можлива, але є дуже складною. Тому довгий час існувала необхідність у створенні стандарту обміну повідомленнями, який би забезпечив взаємодію між різними платформами та службами інтеграції. AMQP вийшов саме для вирішення цієї проблеми.

Протокол розширеної черги повідомлень (AMQP) є важливим протоколом, який широко використовується в останні роки. Він був розроблений Джоном О'Харою з Морганом Чейзом. AMQP може похвалитися чітко визначеним, структурованим набором правил або поведінки для надсилання та отримання повідомлень. Ці правила використовують комбінацію методів, включаючи зберігання та пересилання, публікацію та підписку, один-до-одного, запит / відповідь, кластеризацію, управління транзакціями, завдяки чому протокол став цінним для спілкування між різними операційними системами, платформами програмування, послугами різноманітних інтеграцій та апаратних пристроїв без втрати продуктивності [43].

RabbitMQ - це реалізація стандарту AMQP 0-9-1 із відкритим кодом написаний на Erlang. Він забезпечує підтримку всіх основних операційних систем, а також доступний на Python, Java, Ruby та .Net. RabbitMQ гарно масштабований і надає ряд плагінів, що дозволяють спілкуватися з іншими веб-протоколами, такими як HTTP, XMPP, SMTP та STOMP.

У ході реалізації проекту гостро постало питання з проміжним середовищем для зберігання даних на випадок «відмови» системи. RabbitMQ є відмінним рішенням для побудови SOA (сервісу-орієнтованої архітектури) із розподілом відкладених ресурсномістких завдань. Основна мета RabbitMQ - приймати і віддавати данні. Візуально схема роботи представлена на наступному рисунку (рис. 3.6).

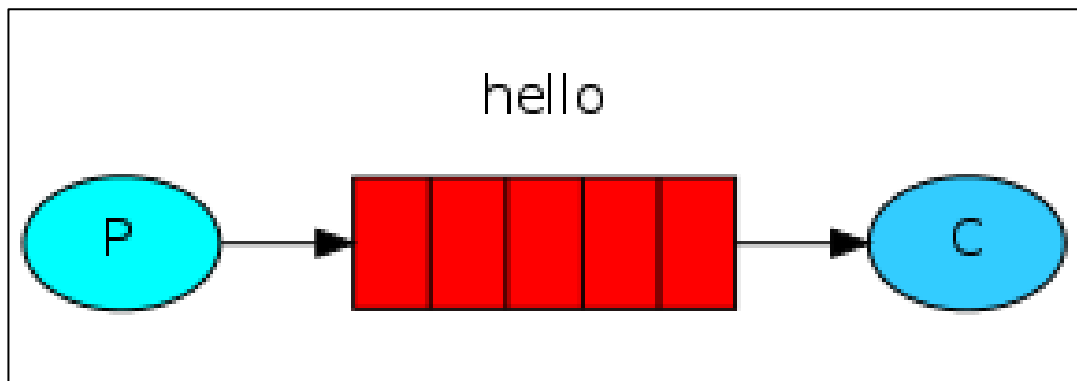


Рисунок 3.6. Умовна схема роботи RabbitMQ.

В обміні повідомленнями використовується наступна термінологія:

- Постачальник - це програма що, надсилає повідомлення. У схемі він представлений колом з літерою «P»;
- Черга – послідовність даних що існує в середовищі RabbitMQ. Незважаючи на те, що дані проходять безпосередньо через Rabbit, усі дані зберігаються виключно у чергах. Черга не має обмежень на кількість даних, що в ній зберігаються. Необмежене число кінцевих пристроїв може відправляти дані у єдину чергу, відповідно й необмежене число «абонентів» може одночасно отримувати дані із однієї черги. У схемі черга позначена набором даних і підписана словом «hello»;
- Абонент – кінцевий пристрій чи програмне забезпечення, що отримує повідомлення. Абонент завжди «чекає» повідомлення. У схемі абонент позначений колом з літерою «C».

Більш детально з візуалізацією роботи брокера повідомлень можна ознайомитись на наступному рисунку (рис. 3.7).

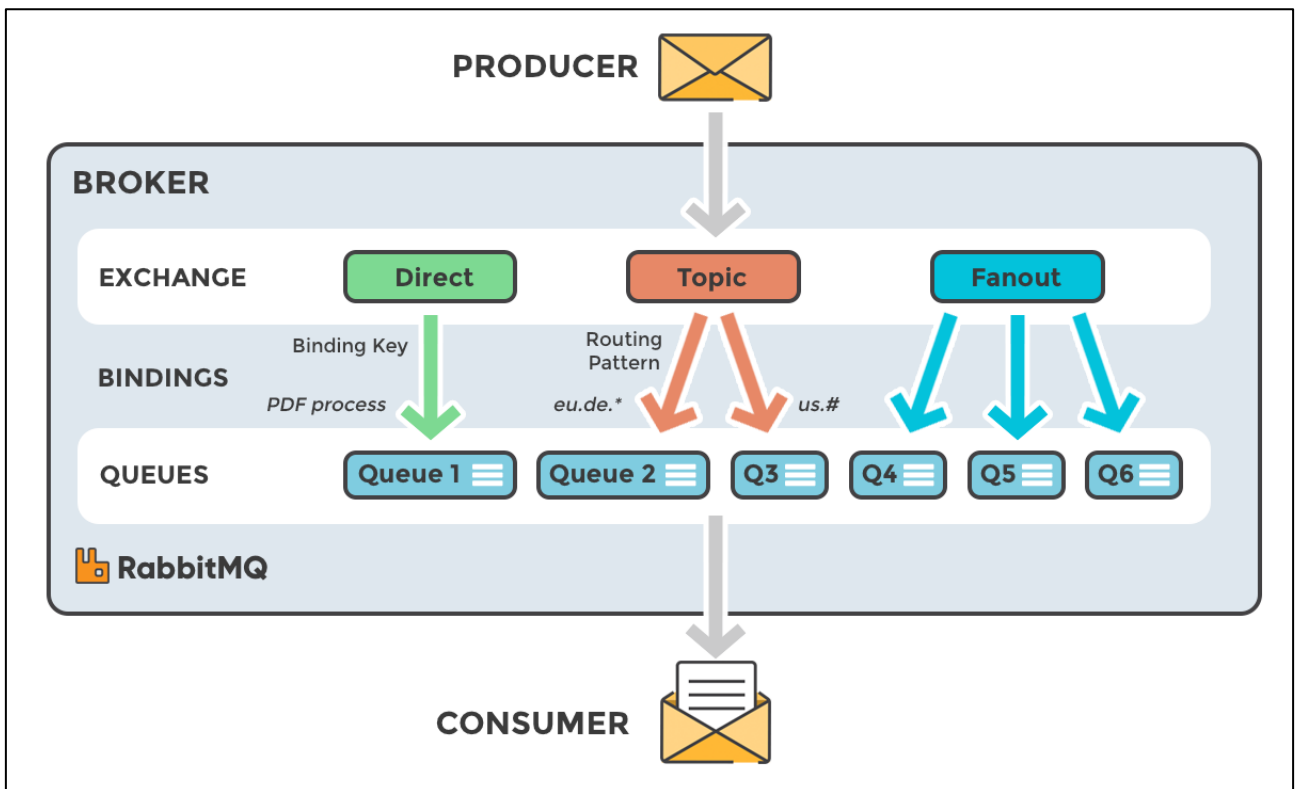


Рисунок 3.7. Схема роботи RabbitMQ.

Зберігання повідомлень у RabbitMQ від усіх поодиноких мікросистем які передають оброблені данні для оптимізації трафіку у глобальному центрі підвищить відмовостійкість запропонованої системи.

3.4. Архітектура проекту оптимізації трафіку на базі IoT-рішення

Розглянемо архітектуру системи оптимізації трафіку на базі IoT-рішення.

Система має реалізувати зв'язок між:

- окремими модулями оптимізації трафіку та світлофорами;
- сусідніми модулями оптимізації трафіку;
- модулями оптимізації трафіку і брокером повідомлень – RabbitMQ;
- брокером повідомлень і глобальним центром обробки даних;
- глобальним центром обробки даних і кожним окремим модулем оптимізації.

Запропонована архітектура системи трьох шарова, і включає в себе три доменні рівні, що являють собою доменний рівень модулів оптимізації трафіку

та світлофорів, мережевий рівень у складі брокера повідомлень — RabbitMQ, та прикладного рівня — глобального центру обробки даних. Архітектура системи оптимізації трафіку на базі IoT-рішення (рис. 3.8) зображена на наступному рисунку.

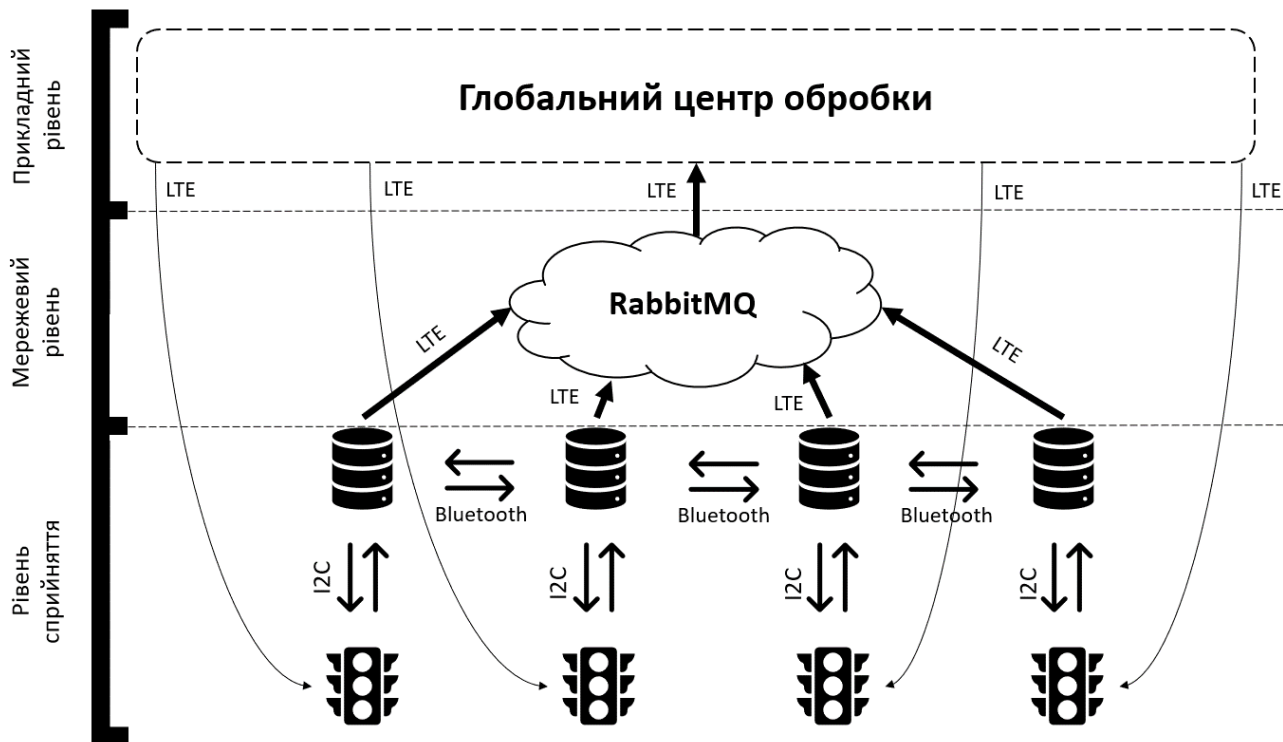


Рисунок 3.8. Архітурне рішення системи оптимізації трафіку.

Рівень сприйняття складається з великої кількості модулів оптимізації трафіку, що розташовані як близько так і на великій відстані один від одного. Кожен модулів оптимізації трафіку складається з датчиків Холла – джерел, що фіксують рух транспортних засобів. Датчики фіксують збурення магнітного поля і передають вимірні дані для подальшої обробки на мікроконтролер Arduino Pro Mini. Мікроконтролер опрацьовує отримані дані і на базі алгоритму оптимізації приймає рішення (керує) підпорядкованим йому світлофором.

Всі зчитані дані з «рівня сприйняття» за допомогою технології LTE надсилаються у брокер повідомлень — RabbitMQ. Цей етап необхідний для підвищення відмовостійкості системи.

RabbitMQ з мережевого рівня передає дані на прикладний рівень у глобальний центр обробки даних. Центр обробки даних використовує їх для

аналізу стану завантаженості автомагістралей міста та побудові глобальної «схеми оптимізації». Відповідно до результату оптимізації трафіку мегаполіса, центр обробки даних надсилає сигнал на мікроконтролер у певній частині міста для управління світлофорним режимом і як результат оптимізації трафіку на транспортних магістралях міста.

3.5. Аналіз захищеності інформації та протоколів для її кіберзахисту

Для забезпечення захищеності інформації під час її передачі між проміжними системами було вирішено використовувати AES шифрування у поєднанні із хеш-функціями [44].

Цей алгоритм перетворює один 128-бітний блок в інший, використовуючи секретний ключ який потрібен для такого перетворення. Для розшифровки отриманого 128-бітного блоку використовують друге перетворення з тим же ключем. Виглядає це так:

```
cipher = encrypt(block, key) // шифруємо block із допомогою key  
block = decrypt(cipher, key) // розшифровуємо cipher із допомогою key
```

Розмір блоку завжди дорівнює 128 біт. Розмір ключа також має фіксоване значення. Щоб зашифрувати довільний текст будь-яким паролем необхідно слідувати наступному алгоритму:

- отримати хеш від пароля;
- перетворити хеш в ключ за правилами описаним у стандарті AES;
- розбити текст на блоки по 128 біт;
- зашифрувати кожен блок функцією cipher.

Це можна записати так:

```
hash = md5(password) // MD5 хеш має довжину 128 біт
key = keyexpansion(hash) // перетворюємо хеш в ключ
blocks = split(text, 16) // розбиваємо текст на блоки по 16 байт

for (i = 0; i < blocks.length; i++)
  cipher[i] = encrypt(blocks[i], key)
```

Щоб розшифрувати масив блоків cipher потрібно застосувати до кожного блоку decrypt:

```
hash = md5(password)
key = keyexpansion(hash)
for (i = 0; i < cipher.length; i++)
  blocks[i] = decrypt(cipher[i], key)

text = merge(blocks) // соединить все блоки в одну строку
```

Звичайно, довжина тексту може бути не кратна 128 бітам. У таких випадках можна доповнити текст нулями до потрібної довжини, а в зашифровані дані додати кілька байт з зашифрованим розміром оригінального тексту.

AES активно використовує так зване кінцеве поле GF (2^8). Поле GF (2^8) це числа 0..255 для яких визначили особливі значення операцій множення і складання. Візьмемо будь-яке число з цього поля і представимо його у вигляді восьми бітів: $a = a_7a_6a_5a_4a_3a_2a_1a_0$. Таким чином представимо й число b . Додавання a і b це відома побітова операція XOR за формулою 3.1:

$$a + b = a \text{ xor } b, \quad (3.1)$$

У додавання досить прості властивості за формулою 3.2:

$$\begin{aligned}
 a + a &= 0 \\
 -a &= 0 - a = a \\
 a - b &= a + (-b) = a + b,
 \end{aligned}
 \tag{3.2}$$

Властивості множення складніші. Запишемо многочлени з коефіцієнтами з бітів цих чисел за формулою 3.3:

$$\begin{aligned}
 p &= a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \\
 q &= b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0,
 \end{aligned}
 \tag{3.3}$$

Тепер перемножимо многочлени і знайдемо залишок від ділення на m за формулою 3.4:

$$\begin{aligned}
 m &= x^8 + x^4 + x^3 + x + 1 \\
 r &= pq \text{ mod } (m),
 \end{aligned}
 \tag{3.4}$$

Чому обрано саме такий m ? У цього многочлена є тільки два дільника-многочлена на яких він ділиться без залишку: одиниця і він сам. За аналогією з простими числами, многочлен m «простий». Знаходити залишок від ділення можна також як для звичайних чисел: для цього достатньо вміти множити, додавати і віднімати многочлени, причому додавання і віднімання працюють за правилами $GF(2^8)$, тобто додавання і віднімання многочленів це XOR між кожною парою коефіцієнтів.

Представимо многочлен r у наступному вигляді за формулою 3.5:

$$r = r_7x^7 + r_6x^6 + r_5x^5 + r_4x^4 + r_3x^3 + r_2x^2 + r_1x + r_0, \tag{3.5}$$

Його 8 коефіцієнтів і є 8-бітове число з поля $GF(2^8)$ і це число називається добутком $a * b$. На відміну від складання, множення можна знайти парою простих побітових операцій. Однак множення на довільний многочлен в поле $GF(2^8)$ можна звести до множення на многочлен x , а помножити на x можна декількома побітові операціями, про що піде мова нижче.

Для позначення многочленів в $GF(2^8)$ використовують 16-ковий цифри, наприклад за формулою 3.6:

$$m = x^8 + x^4 + x^3 + x + 1 = 100011011 = 0x011b = \{01\}\{1b\}, \tag{3.6}$$

Помножити на многочлен $x = \{02\}$ в полі $GF(2^8)$ дуже просто. Розглянемо добуток за формулою 3.7:

$$\begin{aligned} xp &= x(a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0) = \\ &= a_7x^8 + a_6x^7 + a_5x^6 + a_4x^5 + a_3x^4 + a_2x^3 + a_1x^2 + a_0x \\ p &= a7a6a5a4a3a2a1a0 \\ xp &= a7a6a5a4a3a2a1a00, \end{aligned} \tag{3.7}$$

Тепер потрібно знайти залишок від ділення на m . Якщо біт $a_7 = 1$, то потрібно один раз відняти m (7). Якщо $a_7 = 0$ то віднімати нічого не потрібно, розглянути це можна за формулою 3.8:

$$\begin{aligned} r &= xp \text{ mod } (m) = xp - m \text{ якщо } a_7 = 1 \\ r &= xp \text{ mod } (m) = xp \text{ якщо } a_7 = 0, \end{aligned} \tag{3.8}$$

Залишилася одна проста операція в поле $GF(2^8)$. У будь-якого байта b , крім нуля, є зворотний байт $a = b^{-1}$ який має властивість $a \cdot b = \{01\}$.

Таблиця SBox представляє собою 256-байтний масив і використовується для заміни одного байта іншим. Щоб дізнатися чому дорівнює елемент SBox [b] потрібно виконати три дії:

1. знайти зворотний байт до b в поле $GF(28)$ (нуль залишити без змін);
2. помножити результат складається з восьми бітів на матрицю 8×8 з 64 бітів;
3. додати $\{63\}$.

В сумі ці три дії дають Афінне перетворення. Побудовану таблицю (рис. 3.9) можна переглянути на наступному рисунку.

63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Рисунок 3.9 Матриця 8*8.

Для шифрування тексту AES застосовує не пароль або хеш від пароля, а так званий «KeyExpansion» що отримується із ключа. Цей розклад можна уявити як $N_r + 1$ матриць розміру $4 \times N_b$. Алгоритм шифрування робить $N_r + 1$ кроків і на кожному кроці він, крім інших дій, бере одну матрицю $4 \times N_b$ з «розкладу» і поелементно додає її до блоку даних.

Алгоритм шифрування отримує на вхід 128-бітний блок даних `input` і розклад ключів `w`, яке виходить після KeyExpansion. 16-байт `input` він записує у вигляді матриці `s` розміру $4 \times N_b$, яка називається станом AES, і потім N_r раз застосовує до цієї матриці 4 перетворення. В кінці він записує матрицю у вигляді масиву і подає його на вихід - це зашифрований блок. Кожне з чотирьох перетворень дуже просте.

- `AddRoundKey` бере з розкладу ключів одну матрицю розміру $4 \times N_b$ і поелементно додає її до матриці стану. Якщо два рази застосувати `AddRoundKey`, то нічого не зміниться, тому перетворення зворотне до `AddRoundKey` це воно саме;
- `SubBytes` замінює кожен елемент матриці стану відповідним елементом таблиці `SBox`: $s_{ij} = \text{SBox}[s_{ij}]$. Перетворення `SubBytes` оборотно. Зворотне до нього значення знаходиться за допомогою таблиці `InvSBox`;
- `ShiftRows` зсуває i -й рядок матриці s на i позицій вліво, починаючи із нуля. Зворотне перетворення `InvShiftRows` зсуває рядки вправо;
- `MixColumns` помножує кожен стовпець матриці s зліва на особливу матрицю розміру 4×4 .

3.6. Висновки до третього розділу

У третьому розділі була описана архітектура проекту оптимізації трафіку на базі IoT-рішення. Запропонована трьох шарова архітектура, що включає в себе три доменні рівні:

- рівень модулів оптимізації трафіку та світлофорів;
- мережевий рівень у складі брокера повідомлень — `RabbitMQ`;
- прикладний рівень — глобального центру обробки даних.

Рівень сприйняття (мікропроцесорна система зональних контролерів), складається з:

- двох давачів Холла;
- мікропроцесорного мікроконтролера `Arduino Pro Mini`;
- `LTE / Bluetooth` модуля;
- джерела живлення.

Як технології для обміну даними було обрано `Bluetooth` – для обміну із системами ближнього діапазону, і `LTE` – для обміну із глобальним центром оптимізації.

В ролі проміжного програмного забезпечення для обміну даними між системами і глобальним центром було обрано і аргументовано використання брокера повідомлень RabbitMQ. Також використання даного програмного забезпечення дозволяє виконувати швидке і якісне масштабування систем моніторингу і підвищує відмовостійкість.

Для забезпечення крипто безпеки системи був обраний і детально описаний алгоритм AES шифрування у поєднанні із використанням хеш-функцій.

РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСНИХ РІШЕНЬ ТА ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРУ

4.1. Реалізація програмної частини проекту в Arduino IDE

У ході виконання магістерської практики було створено макет діючої системи (рис. 4.1) на базі Arduino Pro Mini і датчиків Холла.

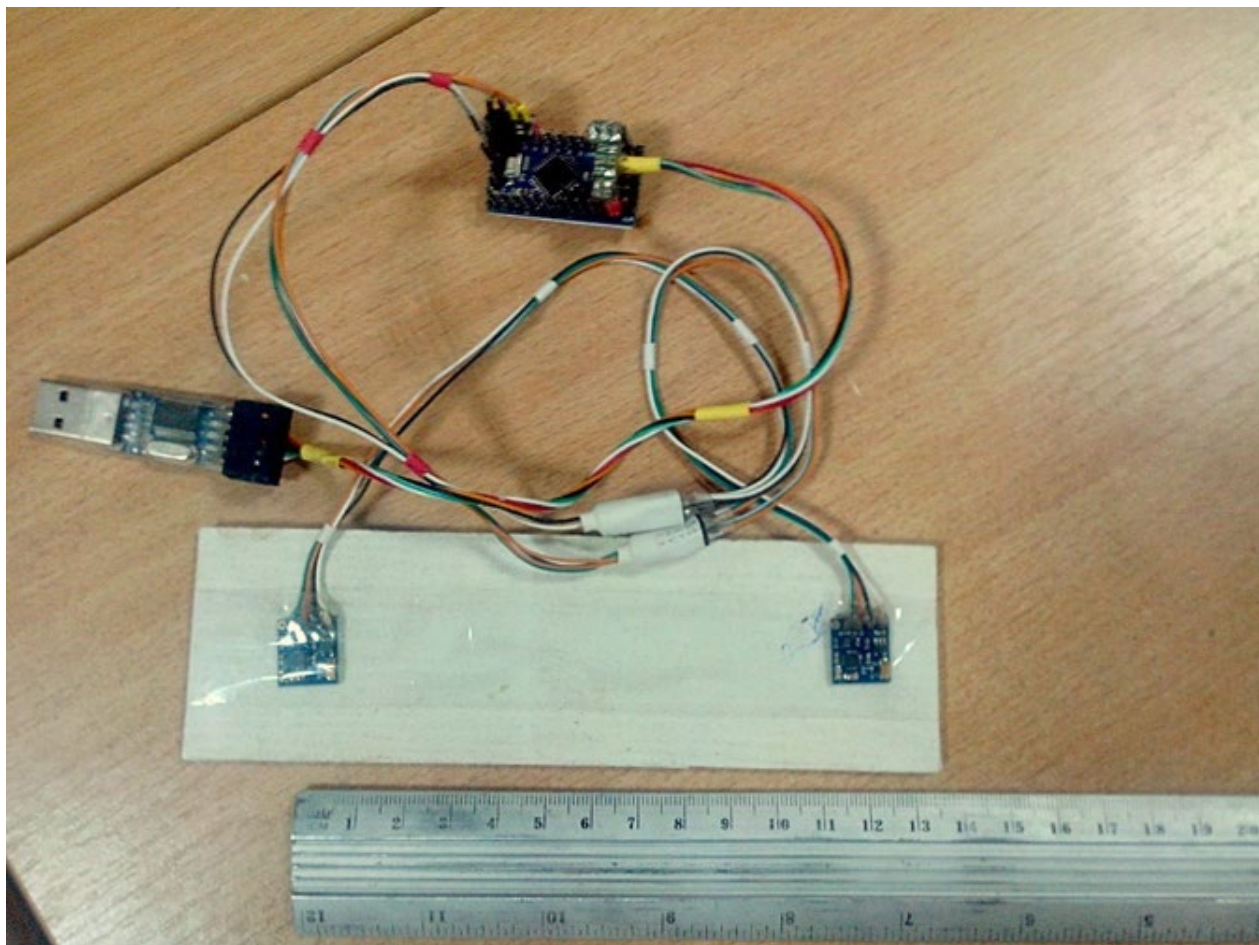


Рисунок 4.1. Макет мікропроцесорної системи.

Для реалізації програмної частини проекту було використано середовище для розробки — Arduino IDE. Дане середовище підтримує написання програмного продукту на C або C++.

Необхідно було написати програму для вимірювання збурень магнітного поля спричиненого рухом авто над датчиками Холла. Вимірюючи затримку між моментами спрацьовування двох давачів планувалося визначати швидкість руху

об'єкта автівок. Реєструючи профіль зміни збурення в часі над одним давачем, і знаючи швидкість руху транспортного засобу, можна визначити довжину об'єкта і почати збирати бібліотеку «магнітних» профілів автівок.

Програма має працювати наступним чином:

- ініціалізація давачів для вимірювання збурень. Для цього підбираються наступні параметри: метод усереднення, частота опитування давачів, значення чутливості;
- вимірювання магнітного поля по координатах X, Y, Z кожного з давачів;
- ініціалізація початкових значень для процедури усереднення і подальшого вилучення постійної складової магнітного поля Землі;

Після цього в нескінченному циклі мікропроцесор виконує наступну послідовність дій:

- розраховується значення магнітного поля з даних отриманих з двох давачів;
- для кожного давача по координатам X, Y, Z обчислюється значення відхилення від поточного середнього значення;
- обчислюється модуль відхилення по координатам X, Y, Z для обох давачів;
- серед відхилень по всім координатам обчислюється максимум для кожного давача. Цей максимум характеризує величину збурення для даного давача.

Процедура обчислення середнього і відхилення від нього використовує наступну рекурентну схему за формулою 4.1:

$$\bar{X}_i = \bar{X}_{i-1} * \left(1 - \frac{1}{N}\right) + \frac{x_i}{N}, \quad (4.1)$$

де x_i — поточне значення вимірюваної величини;

Δx_i — відхилення від середнього значення;

N — довжина інтервалу усереднення.

Як вихідне значення для середнього подається початкове значення вимірної величини — $\bar{X}_0 = x_0$.

Реалізація математичних операцій множення і ділення, вимагають великого завантаження мікропроцесора. Тому було вирішено модифікувати схему за формулою 4.2:

$$\bar{X}_i' = \bar{X}_i * N, \quad (4.2)$$

де $\bar{X}_i = \frac{\bar{X}_i'}{N}$ — істинне середнє значення;

$\bar{X}_i' = \overline{\bar{X}_{i-1}'} - \bar{X}_{i-1+x_i}$ — модифіковане середнє.

Зрозуміло, що це і є добуток вихідного рівняння для середнього помноженого на N за формулою 4.3:

$$\bar{X}_i * N = \left(\bar{X}_{i-1} * \left(1 - \frac{1}{N} \right) + \frac{x_i}{N} \right) * N = \bar{X}_{i-1} * N - \bar{X}_{i-1} + x_i, \quad (4.3)$$

Якщо в якості N обрати число кратне ступеня двійки, тобто $N = 2^P$, тоді операції ділення і множення на N можна змінити на операції зсуву двійкових розрядів. Зсув вліво — $X \gg P$, зсув вправо — $X \ll P$. Цей алгоритм усереднення і реалізований у програмному коді. Фрагмент даного коду продемонстровано на наступному рисунку (рис. 4.2).

```
//----- Averaging and removing slow trend -----
void initAverge(long* pAvrgN, int x){ // init avarege value
    *pAvrgN=(long)x<<_kAveraging; // AvrgN=x*N; N=2^_kAveraging;
}//Average

int difAverge(long* pAvrgN, int x){
    int cAvrg=*pAvrgN>>_kAveraging; // Avrg=AvrgN/N; N=2^_kAveraging;
    *pAvrgN=*pAvrgN-cAvrg+x; // Avrg=Avrg*(1-1/N)+x/N -> AvrgN=AvrgN-AvrgN/N+x
    return x-cAvrg; // dif=x-Avrg
}//Average

//-----
```

Рисунок 4.2. Усереднення та усунення постійної складової магнітного поля.

Для апробації результатів роботи системи було проведено ряд експериментів — над датчиками був пронесений металевий стрижень, спочатку зліва на право, а потім навпаки.

Вхідні параметри експерименту:

1) Відстань між давачами Холла — 10 см.
2) Над давачами на висоті 10-15 см переміщається сталевий стрижень, орієнтований перпендикулярно напрямку переміщення. Діаметр стрижня 5 мм, довжина 15 см.

3) Темп опитування давачів — 20 мс. Стрижень проноситься над давачами тричі: двічі зліва направо і один раз справа наліво. Результати експериментів наведено на наступному рисунку (рис. 4.3).

По наведеному фрагменту можна побачити, що на першому і третьому фрагменту графіку давач S0 спрацьовує раніше, ніж давач S1 — це свідчить про рух зліва на право. На центральному фрагменті графіку видно, що давач S1 спрацьовує раніше, ніж давач S0 — рух справа наліво.

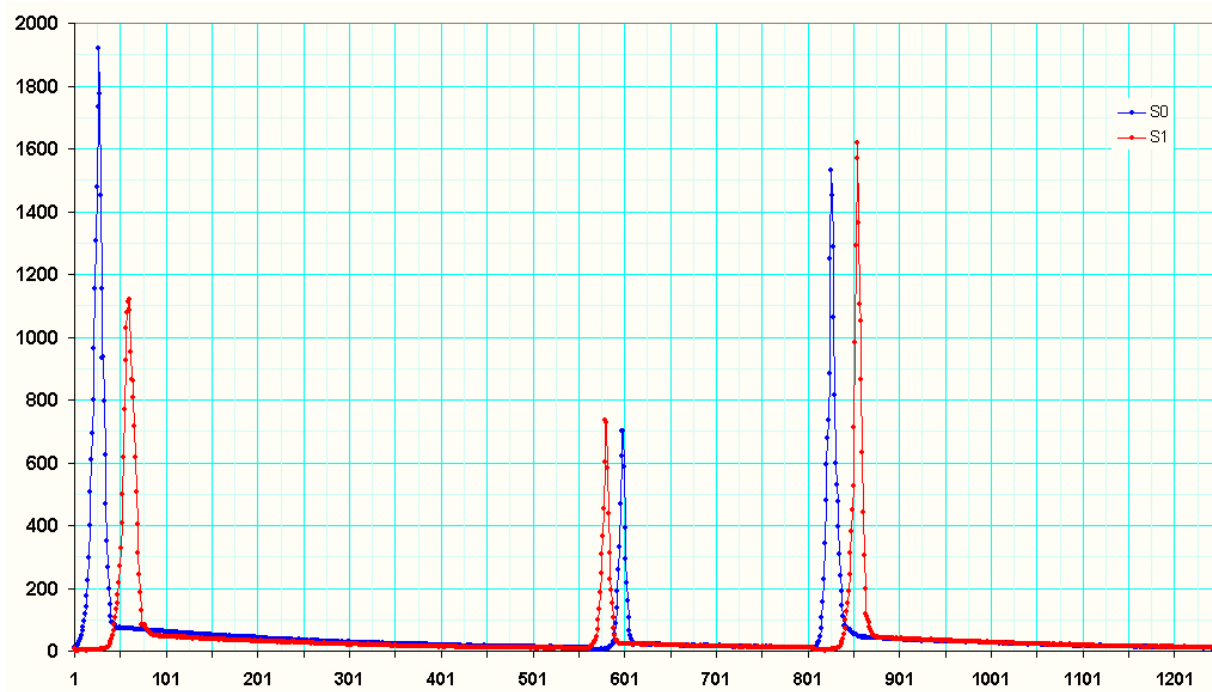


Рисунок 4.3. Результати трьох вимірів збурень магнітного поля.

Розглянемо кожен із фрагментів реєстрації збурень магнітного поля детальніше, перший фрагмент наведено на наступному рисунку (рис. 4.4).

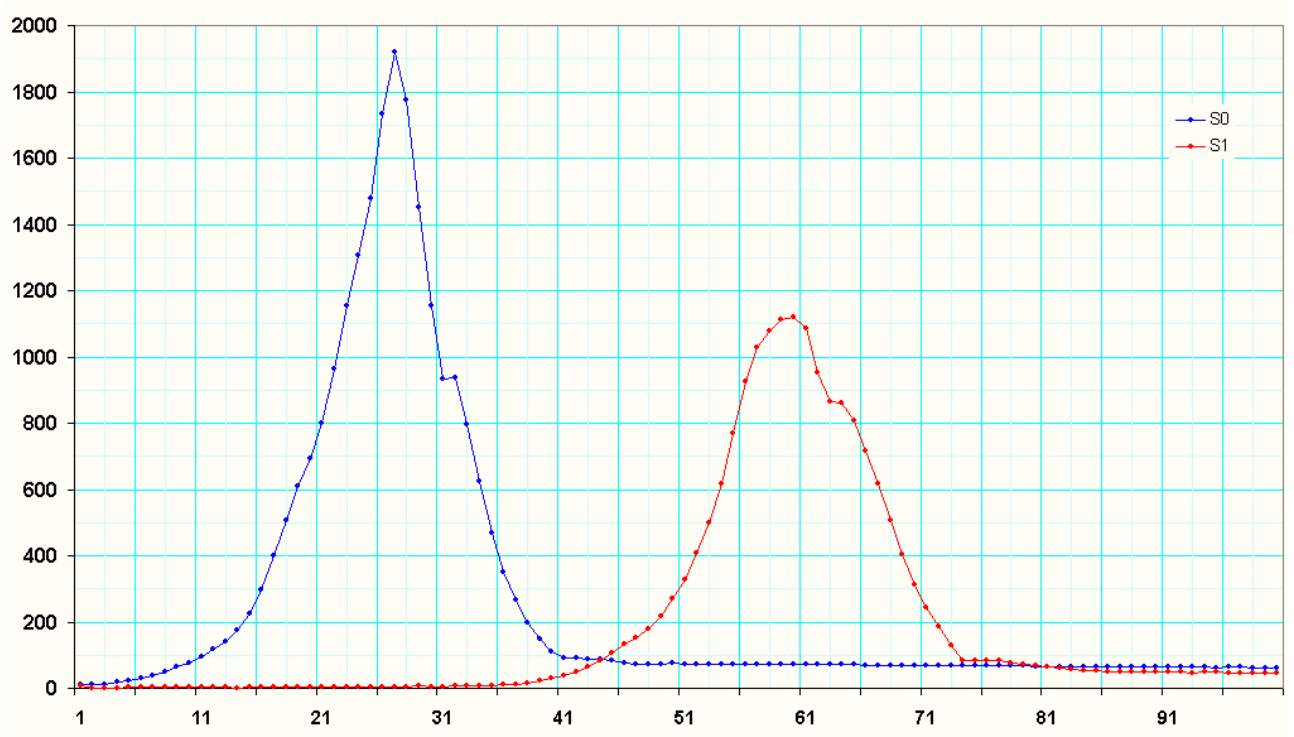


Рисунок 4.4. Перший фрагмент фіксації збурень магнітного поля.

Відстань між збуреннями двох давачів приблизно 35 відділків. Відповідно ми можемо розрахувати час між цими подіями і швидкість руху авто за формулою 4.4:

$$\begin{aligned} \Delta t &= 35 * 20 \text{ мс} = 700 \text{ мс} = 0.7 \text{ с} \\ \Delta V &= \frac{10 \text{ см}}{0.7 \text{ с}} = 14 \frac{\text{см}}{\text{с}}, \end{aligned} \quad (4.4)$$

Розглянемо другий фрагментів реєстрації збурень магнітного поля він наведений на наступному рисунку (рис. 4.5). Відстань між збуреннями двох давачів приблизно 17 відділків. Відповідно ми можемо розрахувати час між цими подіями і швидкість руху авто за формулою (4.5):

$$\begin{aligned} \Delta t &= 17 * 20 \text{ мс} = 340 \text{ мс} = 0.34 \text{ с} \\ \Delta V &= \frac{10 \text{ см}}{0.34 \text{ с}} = 29 \frac{\text{см}}{\text{с}}, \end{aligned} \quad (4.5)$$

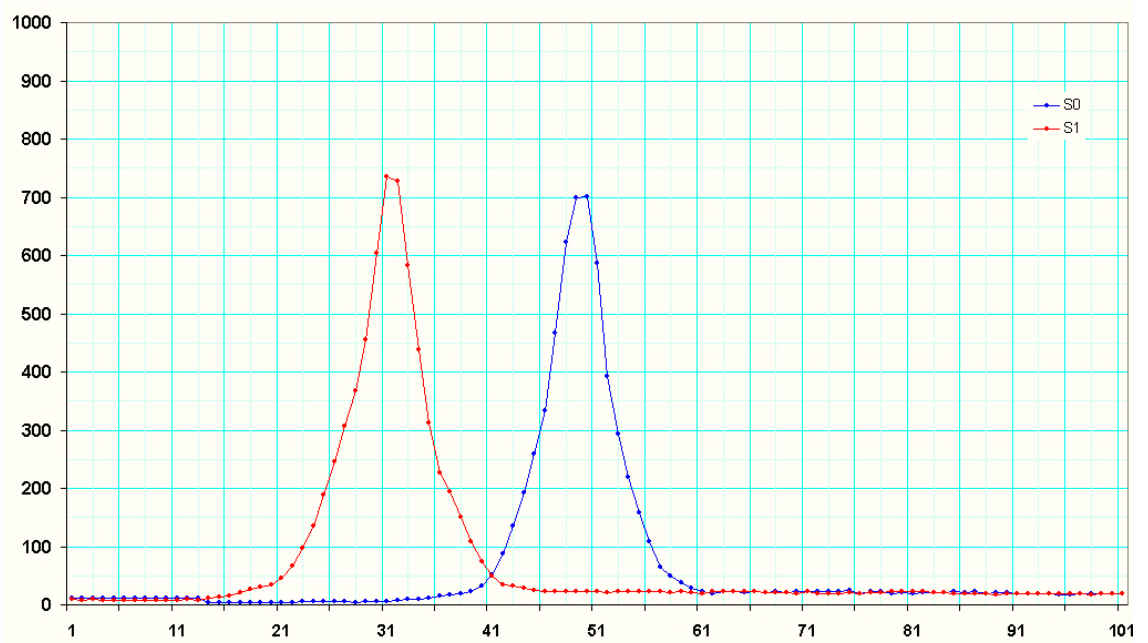


Рисунок 4.5. Другий фрагмент фіксації збурень магнітного поля.

Розглянемо третій фрагментів реєстрації збурень магнітного поля він наведений на наступному рисунку (рис. 4.6). Відстань між збуреннями двох давачів приблизно 30 відділків. Відповідно ми можемо розрахувати час між цими подіями і швидкість руху авто за формулою 4.6:

$$\Delta t = 30 * 20 \text{ мс} = 600 \text{ мс} = 0.6 \text{ с}$$

(4.6)

$$\Delta V = \frac{10 \text{ см}}{0.6 \text{ с}} = 17 \frac{\text{см}}{\text{с}},$$

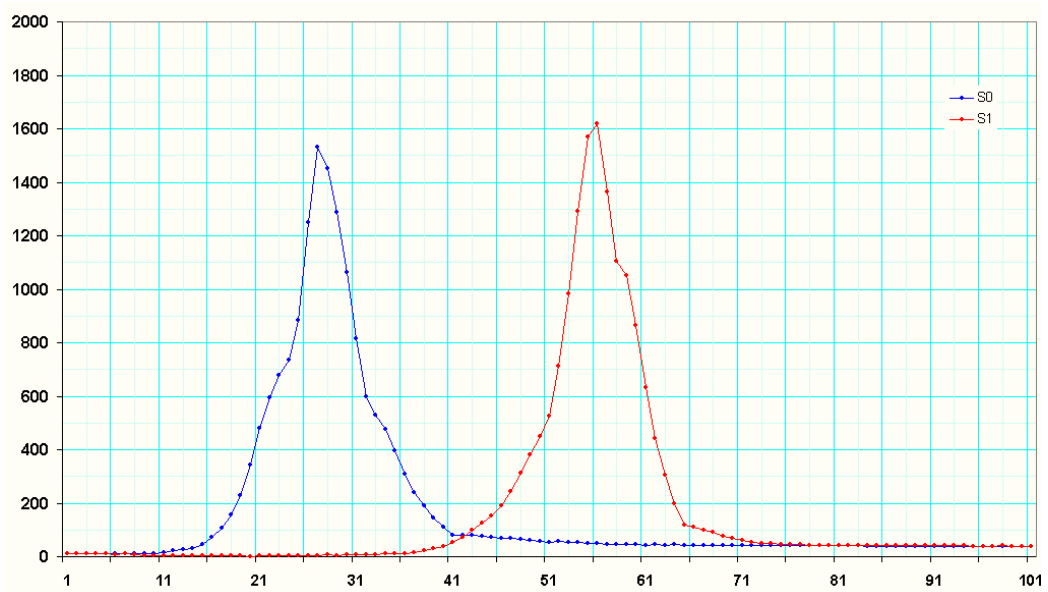


Рисунок 4.6. Третій фрагмент фіксації збурень магнітного поля.

Враховуючі успішність підтвердження гіпотези, щодо вимірів магнітного поля Землі за рахунок датчиків Холла і відмінних експериментальних результатів було вирішено створити візуалізацію збурень магнітного поля у середовищі — LabVIEW.

4.2. Реалізація візуалізації проекту в LabVIEW

Для візуалізації отриманих в ході експериментальної перевірки результатів вимірювання збурень магнітного поля було вирішено обрати графічне середовище для розробки — LabVIEW.

Дане середовище зручно використовувати для швидкого і якісного опрацювання зчитаних даних. LabVIEW надає можливість будувати широкий сект вимірювальної техніки з використанням об'єктно-орієнтовного способу програмування. Більш того, платформа окрім зчитування даних і виконання проміжних розрахунків, дозволяє одночасно будувати візуалізацію розрахованих результатів!

Для обраної системи візуалізації отриманих даних було висунуто чіткі вимоги, а саме реалізувати:

- реалізувати можливість мануального керування вимірюванням швидкості (вмикання/вимикання);
- реалізувати запис зчитаних даних у файл;
- логування можливих помилок;
- реалізації кнопки «STOP» для зупинки програми;
- побудова графіку виміру збурень магнітного поля;
- реалізація індикації напрямку руху авто;
- відображення швидкість руху транспортного засобу.

Так як середовище LabVIEW дозволяє програмувати методом умовного «перетягування» елементів програмного коду реалізація проекту розділяється на два етапи:

- створення програмного продукту;
- створення лицьової панелі.

Пояснення основних елементів представлених на наступному рисунку (рис. 4.7), що відповідає backend програмного продукту:

1. інтерфейс вибору COM-порту підключення Arduino;
2. ініціалізація масиву нулів для запису значень під час збурень;
3. відображення даних із датчиків у виді графіку збурень магнітного поля;
4. реалізація циклу WHILE;
5. реалізація структури «Case»:
 - 5.1. Arduino підключений і готова до проведення вимірів;
 - 5.2. виведення помилки, якщо Arduino не підключена.
6. створення вікна для вказання шлях до файлу запису;
7. перемикач статусу запису у файл;
8. створення заголовків-підписів для стовбців в Excel;
9. реалізація обробника помилок.

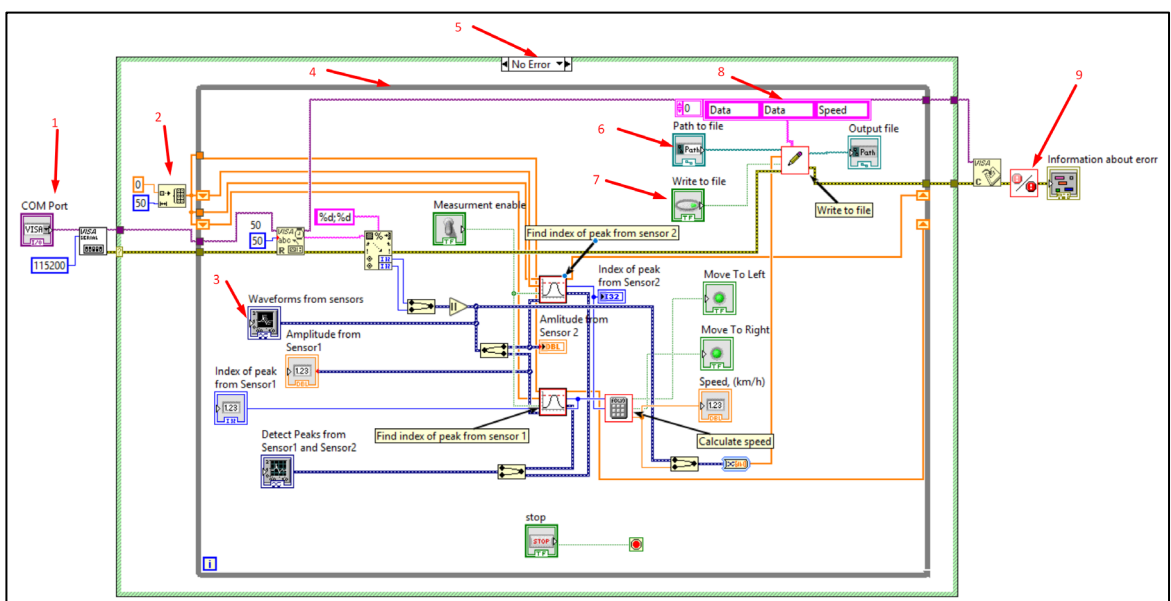


Рисунок 4.7. Backend програмного продукту LabVIEW.

Після реалізації «backend» у програмного продукту LabVIEW перейдемо до створення лицьової панелі, пояснимо елементи зображені на наступному рисунку (рис. 4.8):

1. інтерфейс вибору COM-порту підключення Arduino;
2. тумблер для керування вимірюванням швидкості;
3. тумблер керування записом у файл;
4. вікно для вказання шляху до файлу запису;
5. файл у який відбудеться запис;
6. інтерфейс відображення інформація про помилки;
7. кнопка «STOP», для зупинки перебігу вимірювань;
8. інтерфейс візуалізації вимірів збурень магнітного поля обома давачами;
9. індикатори напрямку руху транспортного засобу;
10. інтерфейс відображення швидкості руху транспортного засобу;
11. інтерфейс візуалізації піків збурень магнітного поля.

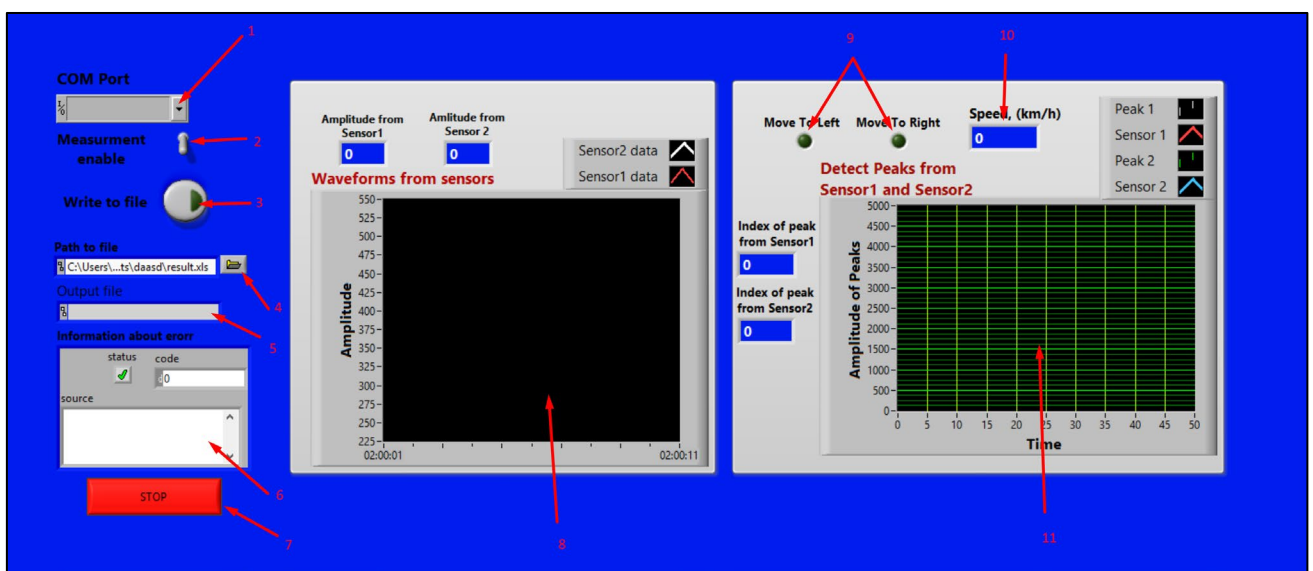


Рисунок 4.8. Лицьова панель програмного продукту LabVIEW.

4.3. Висновки до четвертого розділу

В ході виконня даного розділу на базі Arduino IDE створено програмний код для мікроконтролера Arduino Pro Mini, яке:

- ініціалізує цифровий інтерфейс для двох сенсорів;
- проводить вимірювання магнітного поля по трьох координатах кожного з датчиків;
- по отриманим даним відбувається розрахунок швидкості і напрямку руху транспортного засобу.

Було розроблено та побудовано лабораторний стенд, проведено серію експериментів, а також реалізовано візуалізацію результатів вимірів збурень магнітного поля у середовищі LabVIEW. Під час виконання роботи, досліджено залежність збурення магнітного поля від впливу металічних об'єктів у середовищі LabVIEW, залежність була відображена у вигляді графіку який змінюється у «живому режимі». Отримана характеристика є лінійною, тому що зміна магнітного поля відбувається на певну величину і тримається на цьому рівні.

ВИСНОВКИ

В магістерському проекті вирішувалось питання оптимізації трафіку міського дорожнього руху за допомогою IoT. Для вирішення цього питання було спроектовано систему, яка дозволила збільшити якість моніторингу та допомогла оптимізувати керування потоками транспортних засобів.

В ході написання роботи були вирішені наступні задачі та отримані такі результати:

1. В результаті проведеного аналізу джерел літератури та патентного пошуку були зроблені висновки про необхідність вдосконалення існуючих систем оптимізації трафіку. Існуючі на даний момент системи погано справляються з фіксацією руху транспортних засобів під час багаторядного руху та паркуванні транспортних засобів на дорожніх смугах, або є доволі дорогими. «Вузьким» місцем таких систем є саме давачі інформації, завдяки яким і відбувається фіксація руху автомобілів. Для розроблення нової системи оптимізації було запропоновано використання давачів Холла – чутливих магнітних сенсорів. На базі давачів розроблено алгоритм визначення фіксації напрямку і швидкості руху автомобілів, що надає можливість фіксувати рух автомобілів при багаторядному русі і більш того, при переулаштуванні автівок поміж смугами.

2. На підставі основних функцій та вимог до системи, що проектується, була запропонована схема зонального контролера, до складу якого ввійшли:

- блок зчитування збурень магнітного поля;
- блок обробки (оптимізації) зчитаних даних;
- блок керування світлофорними режимами;
- блок обміну інформацією із навколишніми системами.

Вході виконання магістерської роботи було запропоновано використання чутливих магнітних сенсорів на ефекті Холла в ролі давачів сигналу системи оптимізації трафіку. Також було обґрунтовано використання мікроконтролера

Arduino Pro Mini для виконання операцій по оптимізації дорожнього трафіку шляхом керування світлофорних режимів. Для візуалізації працездатності системи було розроблено інтерфейс фіксації збурень магнітного поля на базі середовища об'єктно-орієнтованого програмування - LabVIEW.

3. Для створення архітектури проекту та проведення експериментальних досліджень щодо обраного методу визначення інтенсивності руху, було створено схему електричну принципіву з використанням Arduino Pro Mini, давачів Холла і модулів передачі даних у зовнішні системи.

В якості технології обміну даними із системами ближнього діапазону було обрано Bluetooth, а для обміну із глобальним центром оптимізації — LTE.

Для забезпечення крипто безпеки системи був обраний і детально описаний алгоритм шифрування AES у поєднанні із використанням хеш-функцій.

4. На етапі створення програмної реалізації інтерфейсних рішень та програмування мікроконтролера було зроблено:

- на базі Arduino IDE написано алгоритми обрахунків швидкості і напрямку руху автомобілів і наведено покроковий процес виконання програми обробки даних;
- візуалізовано збурення магнітного поля Землі феромагнітними частинами, що знаходяться у складі транспортних засобів на базі середовища об'єктно-орієнтованого програмування LabVIEW та описано поблочне виконання об'єктно-орієнтованого коду.

У ході відлагодження системи та проведення лабораторних експериментів було доведено, що обраний метод фіксації руху транспортних засобів, заснований на давачах Холла, та розроблений лабораторний зразок моделі системи оптимізації трафіку, дозволяють визначати швидкість і напрям руху транспортного засобу.

Завдання на магістерське проектування виконано в повному обсязі.

Практична апробація результатів була зроблена у вигляді участі у двох конференціях:

3. Myroshnychenko Y. Road traffic optimization by IoT [Електронний ресурс] / Y. Myroshnychenko, S. Paliy // Information technology & Interactions (Satellite)». – 2020. – Режим доступу до ресурсу: http://iti.fit.univ.kiev.ua/wp-content/uploads/ITI_Satellite_2020_Conference-Proceedings.pdf. [45]

4. Мирошніченко Є.М. Оптимізація руху автотранспорту з використанням розподіленої системи збору даних на базі датчиків Холла : наукова публікація. XVII міжнародна науково-технічна конференція «Проблеми інформатизації», 2021 р. [46]

СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Статистична інформація: Транспорт [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ukrstat.gov.ua/> (дата звертання 04.03.2021)
2. Статистичні дані [Електронний ресурс] – Режим доступу до ресурсу: <https://mtu.gov.ua/>. (дата звертання 04.03.2021)
3. Ju R. S. Transportation quarterly / Eno Foundation for ... v.41 [Електронний ресурс] / R. S. Ju, A. R. Cook, T. H. Maze // Westport, Conn. : The Foundation. – 1987. – Режим доступу до ресурсу: <https://goo.su/2scr>. (дата звертання 06.03.2021)
4. Biloschitsky A. A. Optimizatsiya dorozhnogo trafika s ispolzovaniem sredstv IoT [Електронний ресурс] / A. A. Biloschitsky, Y. M. Myroshnychenko // «Molodiy vcheniy» # 5 (81). – 2020. – Режим доступу до ресурсу: <https://goo.su/2Sd2>. (дата звертання 06.03.2021)
5. Gade D, (2019). ICT based smart traffic management system “iSMART” for smart cities. International Journal of Recent Technology and Engineering, 8(3), 3920–3928. Retrieved from: <https://doi.org/10.35940/ijrte.C5137.098319> (дата звертання 06.03.2021)
6. Ahmed S. A. Urban freeway traffic management technology [Електронний ресурс] / Ahmed // Journal of Transportation Engineering / Volume 112 Issue 4 – Режим доступу до ресурсу: <https://goo.su/2scs>. (дата звертання 06.03.2021)
7. Monitoring avtomobilnyih dorog [Електронний ресурс] // ONIL SAPR GIS. – 2012. – Режим доступу до ресурсу: <https://goo.su/2sCu>. (дата звертання 07.03.2021)
8. Sistemyi dorozhnogo monitoringa. Intellektualnaya transportnaya sistema [Електронний ресурс] // МК Grupp. – 2015. – Режим доступу до ресурсу: <https://goo.su/2scV>. (дата звертання 07.03.2021)

9. Stephen W., Railway wheel sensor employing hall effect devices and corresponding method [Електронний ресурс] / W. Stephen – Режим доступу до ресурсу: <https://bit.ly/2JBI0KM> (дата звертання 07.03.2021)

10. Система для автоматизованого контролю дотримання водієм автомобіля правил дорожнього руху та визначення факту виникнення дорожньо-транспортної пригоди [Електронний ресурс] / [О. О. Лук'янова, О. І. Дохов, О. М. Сітенко та ін.] – Режим доступу до ресурсу: <https://bit.ly/2HiZ2Mn>. (дата звертання 10.03.2021)

11. Осіпов С. К. Способ определения скорости движения и координат транспортных средств с последующей их идентификацией и автоматической регистрацией нарушений правил дорожного движения и устройство для его осуществления [Електронний ресурс] / С. К. Осіпов, А. Ю. Малкін, О. Г. Морская – Режим доступу до ресурсу: <https://bit.ly/2JkciST>. (дата звертання 10.03.2021)

12. HOW DOES SURTRAC WORK? Real-time Control for Traffic Management [Електронний ресурс] // Rapid Flow. – 2021. – Режим доступу до ресурсу: <https://www.rapidflowtech.com/surtrac/how-it-works>. (дата звертання 10.03.2021)

13. Vandeman M. J. Is Traffic Signal Synchronization Justifiable? [Електронний ресурс] / Michael J. Vandeman // Imaja. – 1994. – Режим доступу до ресурсу: <http://www.imaja.com/as/environment/mvarticles/TrafficSigSynch1.html>.

14. Traffic Lights Management Using Optimization Tool. // Creativecommons. – 2018. – №238. – С. 1–8.

15. About Hall Sensors [Електронний ресурс] // АКМ. – 2019. – Режим доступу до ресурсу: <https://www.akm.com/eu/en/technology/technical-tutorial/basic-knowledge-magnetic-sensor/hall-sensors/>. (дата звертання 10.03.2021)

16. Datchyky mahnitnoho polia. Novi zastosuvannia i tekhnolohii vymiriuvannia rukhu i strumu [Електронний ресурс] // Есworld. – 2017. – Режим доступу до ресурсу: <https://goo.su/2SCY>. (дата звертання 10.03.2021)

17. Arduino Pro Mini [Електронний ресурс] // doc.arduino.ua. – 2017. – Режим доступу до ресурсу: <https://doc.arduino.ua/ru/hardware/ProMini>. (дата звертання 14.03.2021)
18. ARDUINO VY`MIRYuYeMO MAGNITNE POLE ZEMLI MAGNITOMETROM HMC5883L [Електронний ресурс] // Mikrotik KPI – Режим доступу до ресурсу: <https://goo.su/2SCZ>. (дата звертання 14.03.2021)
19. M. M. Abbas, Z. M. Adam, and D. Gettman, “Development and evaluation of optimal arterial control strategies for oversaturated conditions,” Transportation Research Record, no. 2259, pp.242–252, 2011.
20. G. Abu-Lebdeh and R. F. Benekohal, “Design and evaluation of dynamic traffic management strategies for congested conditions,” Transportation Research A, vol. 37, no. 2, pp. 109–127, 2003.
21. G. Abu-Lebdeh and R. F. Benekohal, “Signal coordination and arterial capacity in oversaturated conditions,” Transportation Research Record, no. 1727, pp. 68–76, 2000.
22. Y. Li, X. Guo, J. Yang, Y. Liu, and S. He, “Mechanism analysis and implementation framework for traffic signal control of oversaturated intersection group,” Journal of Transportation Systems Engineering and Information Technology, vol. 11, no. 4, pp. 28–34, 2011.
23. M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, “Review of road traffic control strategies,” Proceedings of the IEEE, vol. 91, no. 12, pp. 2043–2067, 2003.
24. Transportation Research Board, Highway Capacity Manual, National Academy of Sciences, Washington, DC, USA, 5th edition, 2010.
25. A. G. Sim and K. W. Dobinson, “The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits,” IEEE Transactions on Vehicular Technology, vol. 29, no. 2, pp. 130–137, 1980.

26. R. D. Bretherton and G. T. Bowen, "Recent enhancements to SCOOT-SCOOT version 2.4," in Proceedings of the 3rd International Conference on Road Traffic Control, pp. 95–98, London, UK, May 1990.
27. F. Dion, H. Rakha, and Y.-S. Kang, "Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections," *Transportation Research B*, vol. 38, no. 2, pp. 99–122, 2004.
28. R. W. Denney, L. Head, and K. Spencer, *Signal Timing under Saturated Conditions*, Federal Highway Administration, Washington, DC, USA, 2008.
29. B. B. Park and C. Li, "An analytical approach for estimating the highway capacity manual signalized intersection delay variability," *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 4, pp. 327–333, 2011.
30. F. Zheng and H. Van Zuylen, "Uncertainty and predictability of urban link travel time: delay distribution-based analysis," *Transportation Research Record*, no. 2192, pp. 136–146, 2010.
31. A. Chen, H. Yang, H. K. Lo, and W. H. Tang, "Capacity reliability of a road network: an assessment methodology and numerical results," *Transportation Research B*, vol. 36, no. 3, pp. 225–252, 2002.
32. E. J. Smaglik, D. M. Bullock, D. Gettman, C. M. Day, and H. Premachandra, "Comparison of alternative real-time performance measures for measuring signal phase utilization and identifying oversaturation," *Transportation Research Record*, no. 2259, pp. 123–131, 2011.
33. K. Ghoseiri, F. Szidarovszky, and M. J. Asgharpour, "A multiobjective train scheduling model and solution," *Transportation Research B*, vol. 38, no. 10, pp. 927–952, 2004.
34. J.-H. Kim, J.-H. Han, Y.-H. Kim, S.-H. Choi, and E.-S. Kim, "Preference-based solution selection algorithm for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 20–34, 2012.

35. J. Jie, J. Zeng, C. Han, and Q. Wang, “Knowledge-based cooperative particle swarm optimization,” *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 861–873, 2008.
36. W.-C. Hong, “Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm,” *Neurocomputing*, vol. 74, no. 12-13, pp. 2096–2107, 2011.
37. R. Putha, L. Quadrioglio, and E. Zechman, “Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 1, pp. 14–28, 2012.
38. B. Park, C. Messer, and T. Urbanik II, “Traffic signal optimization program for oversaturated conditions: genetic algorithm approach,” *Transportation Research Record*, no. 1683, pp. 133–142, 1999.
39. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
40. D. Sun, R. F. Benekohal, and S. T. Waller, “Multiobjective traffic signal timing optimization using non-dominated sorting genetic algorithm,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 198–203, Columbus, Ohio, USA, 2003.
41. Internet of Things - Wireless data transfer technologies [Электронный ресурс] // SOS electronics. – 2017. – Режим доступа до ресурсу: <https://www.soselectronic.com/articles/sos-supplier-of-solution/internet-of-things-wireless-data-transfer-technologies-part-2-2043>. (дата звертання 11.04.2021)
42. Sbor i obrabotka dannyih IoT. Stoit li vnedryat i kak eto sdelat effektivno [Электронный ресурс] // DIS Group. – 2018. – Режим доступа до ресурсу: <https://goo.su/2Sd0>. (дата звертання 11.04.2021)
43. RabbitMQ tutorial 1 — Hello World [Электронный ресурс] // Habr. – 2012. – Режим доступа до ресурсу: <https://habr.com/ru/post/149694/>. (дата звертання 11.04.2021)

44. Как устроен AES [Електронний ресурс] // Хабр. – 2011. – Режим доступу до ресурсу: <https://habr.com/ru/post/112733/>. (дата звертання 11.04.2021)

45. Myroshnychenko Y. Road traffic optimization by IoT [Електронний ресурс] / Y. Myroshnychenko, S. Paliy // Information technology & Interactions (Satellite)». – 2020. – Режим доступу до ресурсу: http://iti.fit.univ.kiev.ua/wp-content/uploads/ITI_Satellite_2020_Conference-Proceedings.pdf. (дата звертання 11.04.2021)

46. Мирошниченко Є.М. Оптимізація руху автотранспорту з використанням розподіленої системи збору даних на базі датчиків Холла : наукова публікація. XVII міжнародна науково-технічна конференція «Проблеми інформатизації», 2021 р. (дата звертання 11.04.2021)

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ МІСЬКОГО ДОРОЖНЬОГО
РУХУ ЗА ДОПОМОГОЮ ІoT
Текст програми для мікроконтролеру Arduino Pro Mini

Листів 5

Розробник _____ Єгор МИРОШНИЧЕНКО

Н

Київ, 2021

```

1  /*
2  HMC5883L 2 chanelS
3  Intreface:
4  * SDC_0 - D2
5  * SDA_0 - D3
6  * SDC_1 - D4
7  * SDA_1 - D5
8  * 3.3V
9  * Gnd
10 */
11
12 #define _kAvereging 8 // N=256; N=2^_kAvereging
13 #define _Level 10
14
15 // Sensors pins
16 #define _SDC_0 2
17 #define _SDA_0 3
18 #define _SDC_1 4
19 #define _SDA_1 5
20
21
22 #define _Bods 115200
23 #define _Led 13
24
25 #define _ADDR 0x1E
26 #define _gain 0 // 0-7 0=most sensitive
27 #define _avrg 3 //0-3 3=8 samples
28 #define _rate 6 //0-6 6=75Hz
29 #define _noBias 0 //normal conditions
30 #define _pBias 1 //positive bias field
31 #define _nBias 2 //negative bias field
32 #define _cont 0 //continuouse
33 #define _single 1 //single
34 #define _mDelay 6 //ms
35
36 #define _gPause 0 //ms 0-> 20..21ms
37
38 #include <AES.h>
39 #include <Bluetooth.h>
40 #include <LTE.h>
41 #include <SoftI2cMaster.h>
42
43 SoftI2cMaster ch[2];
44
45 long XavrN[2],YavrN[2],ZavrN[2]; // Avrg*N chanelS 0/1
46 int X[2],Y[2],Z[2]; // Current values chanelS 0/1

```

```

48 void setup() {
49     ch[0].init(_SDC_0,_SDA_0);
50     ch[1].init(_SDC_1,_SDA_1);
51
52     Serial.begin(_Bods);
53     Serial.println();
54     Serial.print("setup");
55     pinMode(_Led,OUTPUT);
56
57     setConditions(ch[0]);
58     setConditions(ch[1]);
59     digitalWrite(_Led,HIGH);
60     delay(100); // to stabilize enveronment conditions
61     digitalWrite(_Led,LOW);
62
63     // save inital values
64     singleRead(ch[0],0);
65     singleRead(ch[1],1);
66     for (int c=0; c<2; c++){
67         initAverege(&XavrN[c],X[c]);
68         initAverege(&YavrN[c],Y[c]);
69         initAverege(&ZavrN[c],Z[c]);
70     }//
71
72 }//setup
73
74 void loop() {
75     digitalWrite(_Led,HIGH);
76     singleRead(ch[0],0);
77     singleRead(ch[1],1);
78     digitalWrite(_Led,LOW);
79
80     for (int c=0; c<2; c++){
81         // remove constant values
82         X[c]=difAverege(&XavrN[c],X[c]);
83         Y[c]=difAverege(&YavrN[c],Y[c]);
84         Z[c]=difAverege(&ZavrN[c],Z[c]);
85
86         X[c]=abs(X[c]); Y[c]=abs(Y[c]); Z[c]=abs(Z[c]);
87         // maximum disturbance
88         if (Y[c]>X[c]) X[c]=Y[c]; // maximum X,Y
89         if (Z[c]>X[c]) X[c]=Z[c]; // maximum X,Z
90     }//
91

```

```

92     if ((X[0]>_Level)||X[1]>_Level)){
93         Serial.print("*");
94         Serial.print(X[0]); //chanel 0
95         Serial.print(" ");
96         Serial.print(X[1]); //chanel 1
97     }//if
98     delay(_gPause);
99 }//loop
100
101 //----- Avereging and removing slow trend
102
103 void initAverege(long* pAvrgN, int x){ // init avarege value
104     *pAvrgN=(long)x<<_kAvereging; // AvrgN=x*N; N=2^_kAvereging;
105 }//Average
106
107 int difAverege(long* pAvrgN, int x){
108     int cAvrg=*pAvrgN>>_kAvereging;
109     // Avrg=AvrgN/N; N=2^_kAvereging;
110     *pAvrgN=*pAvrgN-cAvrg+x;
111     // Avrg=Avrg*(1-1/N)+x/N -> AvrgN=AvrgN-AvrgN/N+x
112     return x-cAvrg; // dif=x-Avrg
113 }//Average
114 //-----
115
116 //----- Compas sensors processing --
117
118 void setConditions(SoftI2cMaster ch){
119     ch.start(_ADDR<<1);
120     ch.write(0x00); //Register A
121     //0AARRRBB - Avereging 1,2,4,8;
122     // Rate Hz 0.75,1.5,3.0,7.5,15(def),30,75; Bias 0,+,-;
123     ch.write(_avrg<<5 | _rate<<2 | _noBias);
124     ch.stop();
125     ch.start(_ADDR<<1);
126     ch.write(0x01); //Register B
127     //GGG00000 - Gain for 1Gauss
128     //1370,1090(def),820,660,440,390,330,230
129     ch.write(_gain<<5); // gain
130     ch.stop();
131     ch.start(_ADDR<<1);
132     ch.write(0x02); //Mode Register
133     //H00000MM - I2C speed normal/high;
134     //Mode 0-continuous, 1-single(def), 2,3-idle
135     ch.write(_single);
136     ch.stop();
137     delay(_mDelay);
138 }//setConditions

```

```

140 void singleRead(SoftI2cMaster ch, char iSensor) {
141     ch.start(_ADDR<<1);
142     ch.write(0x02); //Mode Register
143     ch.write(_single);
144     ch.stop();
145     delay(_mDelay);
146     ch.start(_ADDR<<1);
147     ch.write(0x03); // point to X-register
148     ch.stop();
149     ch.start(_ADDR<<1 | 1); //read
150     X[iSensor]=(int)ch.read(0)<<8 | (int)ch.read(0); //x
151     Z[iSensor]=(int)ch.read(0)<<8 | (int)ch.read(0); //z
152     Y[iSensor]=(int)ch.read(0)<<8 | (int)ch.read(1); //y
153     ch.stop();
154 }//singleRead
155
156

```

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ МІСЬКОГО ДОРОЖНЬОГО
РУХУ ЗА ДОПОМОГОЮ ІoT

Графічні зображення

Листів 11

Розробник

_____ Єгор МИРОШНИЧЕНКО

Н

Київ, 2021

Зміст

Слайд 1 Тема доповіді.....	3
Слайд 2 Вступ	3
Слайд 3 Цілі кваліфікаційної роботи.....	4
Слайд 4 Модель системи оптимізації - «Surtrac»	4
Слайд 5 Порівняння давачів.....	5
Слайд 6 Описання методу фіксації руху.....	5
Слайд 7 Структурна схема.....	6
Слайд 8 Архітектурне рішення.....	6
Слайд 9 Алгоритм оптимізації.....	7
Слайд 10 Реалізація апаратної частини проекту	7
Слайд 11 Експериментальна частина дослідження.....	8
Слайд 12 Візуалізація вимірювань.....	8
Слайд 13 Апробація.....	9
Слайд 14 Висновки.....	9
Слайд 15 Дякую за увагу!.....	10



КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ



ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ МІСЬКОГО ДОРОЖНЬОГО РУХУ ЗА ДОПОМОГОЮ ІОТ

ПІДГОТУВАВ: СТУДЕНТ 2-ГО КУРСУ, ГРУПИ ІРМ-21 — МИРОШНИЧЕНКО ЄГОР МАКСИМОВИЧ

НАУКОВИЙ КЕРІВНИК: КАНДИДАТ ТЕХНІЧНИХ НАУК, ДОЦЕНТ — ПАЛІЙ СЕРГІЙ ВОЛОДИМИРОВИЧ

Київ - 2021

Рисунок Б 1. Слайд 1 Тема доповіді

ВСТУП

- 1. МЕТОЮ МАГІСТЕРСЬКОЇ РОБОТИ** СТАЛО СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОПТИМІЗАЦІЇ МІСЬКОГО ДОРОЖНЬОГО РУХУ ЗА ДОПОМОГОЮ ЗАСОБІВ ІОТ НА БАЗІ ЗБОРУ І ОПРАЦЮВАННЯ ДАНИХ ЗЧИТАНИХ З ЧУТЛИВИХ СЕНСОРІВ МАГНІТНОГО ПОЛЯ.
- 2. ОБ'ЄКТОМ МАГІСТЕРСЬКОЇ РОБОТИ** СТАЛА ІНФОРМАЦІЙНО-АПАРАТНА СИСТЕМА ДЛЯ ВИЗНАЧЕННЯ ІНТЕНСИВНОСТІ РУХУ ТРАНСПОРТНИХ ЗАСОБІВ. ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ НАЛЕЖИТЬ ДО ОБЛАСТІ СИСТЕМ ОПТИМІЗАЦІЇ ДОРОЖНЬОГО РУХУ.
- 3. ПРЕДМЕТ МАГІСТЕРСЬКОЇ РОБОТИ** Є СПОСІБ ФІКСАЦІЇ РУХУ АВТОТРАНСПОРТУ І РОЗРОБКА ВІДПОВІДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ЗЧИТАНИХ ПОКАЗНИКІВ.

Рисунок Б 2. Слайд 2 Вступ

ЦІЛІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

ЦІЛЮ ДАНОЇ РОБОТИ СТАЛО СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ДОРОЖНЬОГО РУХУ НА БАЗІ ОБРОБКИ ДАНИХ, ЗЧИТАНИХ З СЕНСОРІВ МАГНІТНОГО ПОЛЯ, ДЛЯ ПОЛІПШЕННЯ МОНІТОРИНГУ ТРАНСПОРТНОГО РУХУ І КЕРУВАННЯ РУХОМ У МЕГАПОЛІСАХ, РЕГУЛЮВАННЯ ТРИВАЛОСТІ СВІТЛОФОРНИХ ТАКТИВ ТА СВОЄЧАСНОГО ВМИКАННЯ ІНФОРМАЦІЙНИХ ЗНАКІВ.

Для досягнення цілей необхідно:

1. Провести аналіз існуючих систем оптимізації трафіку і даччиків інформації на яких вони базуються;
2. Описати метод використання датчиків Холла в якості даччиків системи;
3. Описати і побудовані логічна і структурна схема запропонованої системи оптимізації трафіку;
4. Розробити архітектурне рішення системи на базі IoT;
5. Розробити програмне забезпечення для системи моніторингу;
6. Розробити експериментальний макет системи;
7. Розробити програмне забезпечення для системи візуалізації даних - LabVIEW;
8. Описати використання проміжної системи передачі даних – RabbitMQ;
9. Описати алгоритм оптимізації трафіку;
10. Виконати апробацію кваліфікаційної роботи.

Рисунок Б 3. Слайд 3 Цілі кваліфікаційної роботи

МОДЕЛЬ СИСТЕМИ ОПТИМІЗАЦІЇ - «SURTRAC»

Алгоритм роботи системи можна описати наступним чином:

1. СИСТЕМА SURTRAC — збирає данні із оточення про стан дорожньої ділянки в реальному часі, за допомогою API інтеграції із існуючою інфраструктурою, яка може включати камери, радари або індукційні ворота.
2. SURTRAC обробляє цю інформацію, а потім щосекунди за допомогою програмного забезпечення створює план оптимізації руху автотрафіку через перехрестя.
3. Наступним кроком SURTRAC надсилає команди контролеру для керування системами інформування водіїв / зміненню світлофорних тактів.
4. Нарешті, SURTRAC передає інформацію про зчитані данні по завантаженості ділянки дороги до сусідніх перехресть, щоб вони могли включити цю інформацію у свої плани по оптимізації, що дозволяє виконувати як автономне керування трафіком на ділянках доріг, так і виконувати оптимізацію на рівні мережі SURTRAC, в залежності від умов дорожнього руху.



Рисунок Б 4. Слайд 4 Модель системи оптимізації - «Surtrac»

ПОРІВНЯННЯ ДАВАЧІВ

Існуючі рішення	Оптичні датчики	Ультразвукові датчики	Індуктивні ворота	Відеоспостереження	Датчики Холла
Проблеми					
Багаторядний рух	Схема з перетином променя (при багаторядному русі неможливо виконати кількісну оцінку, визначити напрямок руху і швидкість автомобіля).	Схема з перетином променя (при багаторядному русі неможливо виконати кількісну оцінку, визначити напрямок руху і швидкість автомобіля).	Недоцільне і дороге встановлення воріт при багаторядному русі (неможливо виконати кількісну оцінку, визначити напрямок руху і швидкість автомобіля).	Проблема вирішується збільшенням кількості камер, або особливим налаштуванням програмного забезпечення, що відповідно впливає на вартість системи в цілому.	Проблема фіксації руху транспортних засобів при багаторядному русі вирішується за рахунок вимірів збурень магнітного поля Землі по трьох координатах (XYZ).
Бруд	Відбиття променя від днища автотранспорту може не відбуватися через бруд і наявність масляних покриттів на ньому.	Неможливо захистити активний елемент від бруду, тому що він повинен завжди бути відкритим.	Ніяк не впливає на функціонування магнітних воріт.	Є вірогідність забруднення камер спостереження.	Ніяк не впливає на функціонування датчиків Холла.
Вартість	65\$	60\$	180\$	200\$	6\$

Рисунок Б 5. Слайд 5 Порівняння датчиків

ОПИСАННЯ МЕТОДУ ФІКСАЦІЇ РУХУ

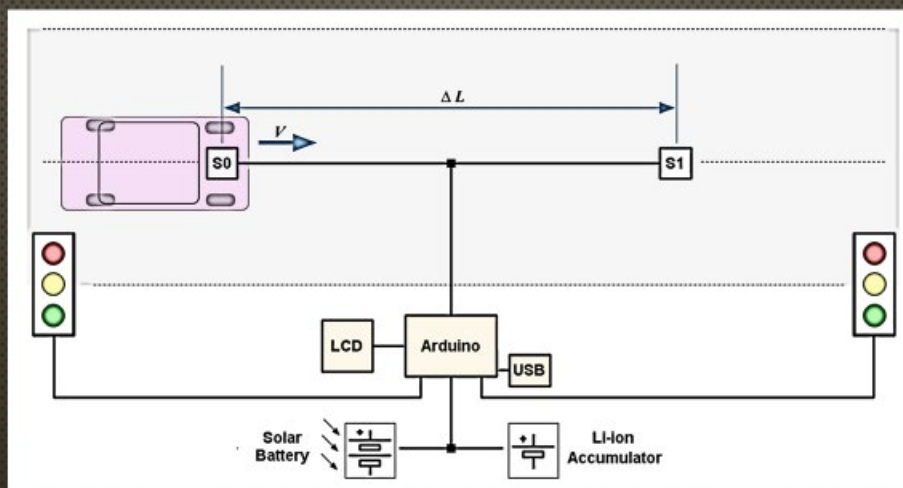


Рисунок Б 6. Слайд 6 Описання методу фіксації руху

СТРУКТУРНА СХЕМА

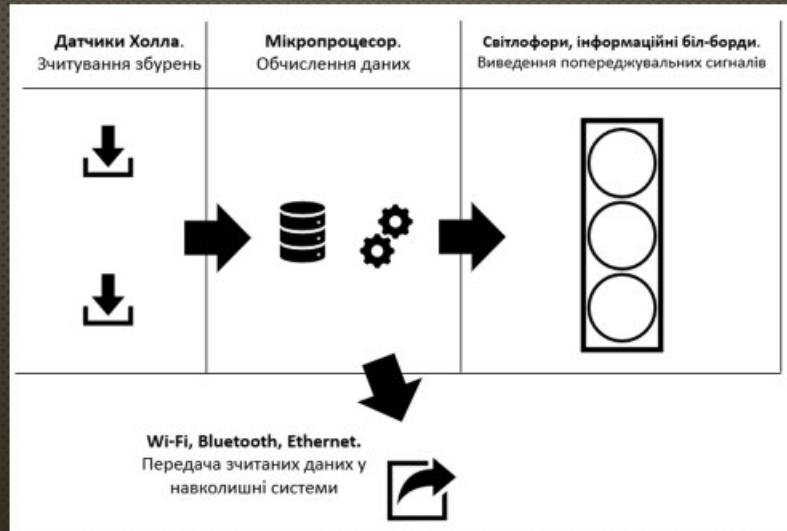


Рисунок Б 7. Слайд 7 Структурна схема

АРХІТЕКТУРНЕ РІШЕННЯ

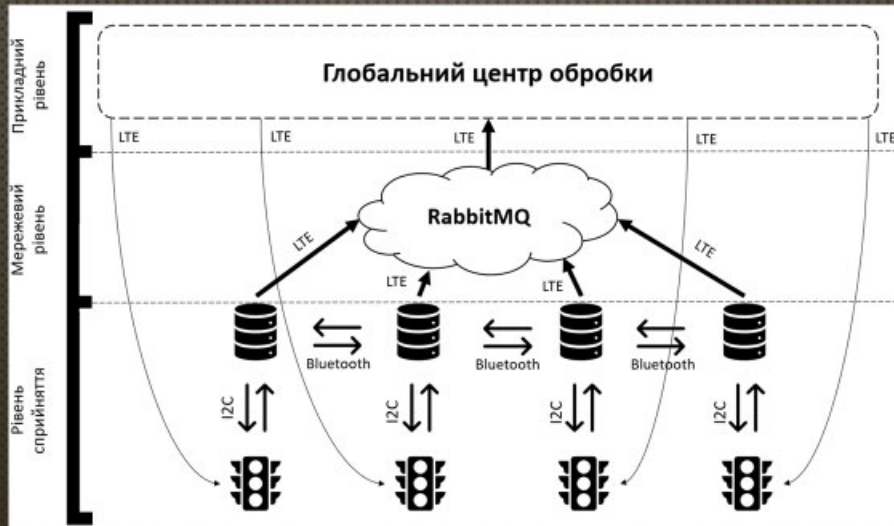


Рисунок Б 8. Слайд 8 Архітектурне рішення

АЛГОРИТМ ОПТИМІЗАЦІЇ

В ЯКОСТІ АЛГОРИТМУ ОПТИМІЗАЦІЇ ДОРОЖНЬОГО ТРАФІКУ БУВ ОБРАНИЙ ОДИН ІЗ НАЙБІЛЬШ ЕФЕКТИВНИХ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ БАГАТО ОБ'ЄКТНОЇ ОПТИМІЗАЦІЇ — ГЕНЕТИЧНИЙ АЛГОРИТМ З НЕДОМІНУЮЧИМ СОРТУВАННЯМ (NSGA-II АБО NON-DOMINATED SORTING GENETIC ALGORITHM).

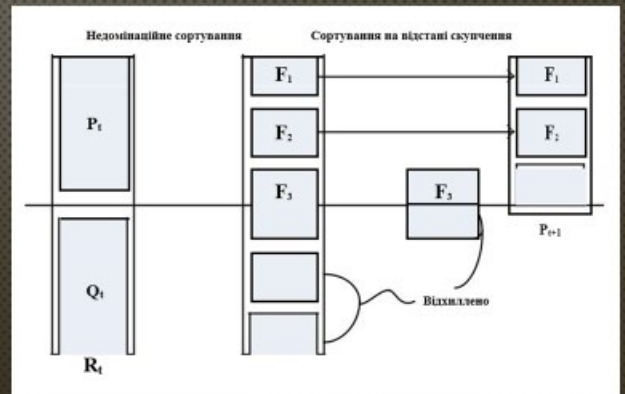


Рисунок Б 9. Слайд 9 Алгоритм оптимізації

РЕАЛІЗАЦІЯ АПАРАТНОЇ ЧАСТИНИ ПРОЕКТУ

У ХОДІ РЕАЛІЗАЦІЇ АПАРАТНОЇ ЧАСТИНИ ПРОЕКТУ БУЛО ЗІБРАНО МАКЕТ НА БАЗІ МІКРОПРОЦЕСОРНОЇ ПЛАТИ ARDUINO PRO MINI ТА ДВОХ ДАВАЧІВ ХОЛЛА.

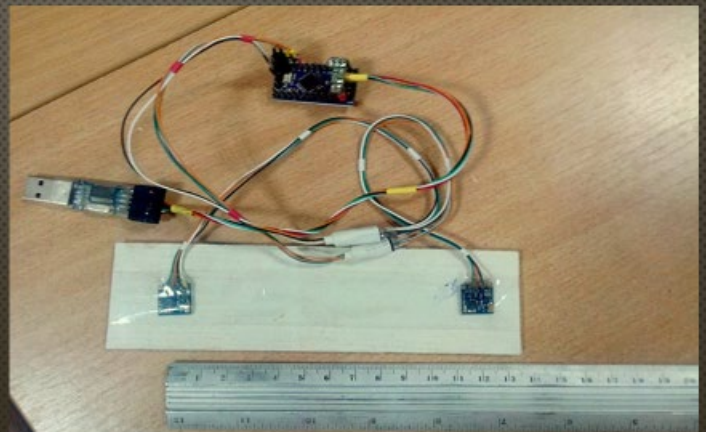


Рисунок Б 10. Слайд 10 Реалізація апаратної частини проекту

ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ДОСЛІДЖЕННЯ

Для проведення лабораторних експериментів над макетом було пронесено металевий стрижень. На побудованому в зафіксованих даних графіку ми можемо побачити збурення на обох давачах.

Відстань між збуреннями 35 поділок, час опитування 20 мс. Можна розрахувати наступне:

$$\Delta t = 35 * 20 \text{ мс} = 700 \text{ мс} = 0.7 \text{ с}$$
$$\Delta V = \frac{10 \text{ см}}{0.7 \text{ с}} = 14 \frac{\text{см}}{\text{с}}$$

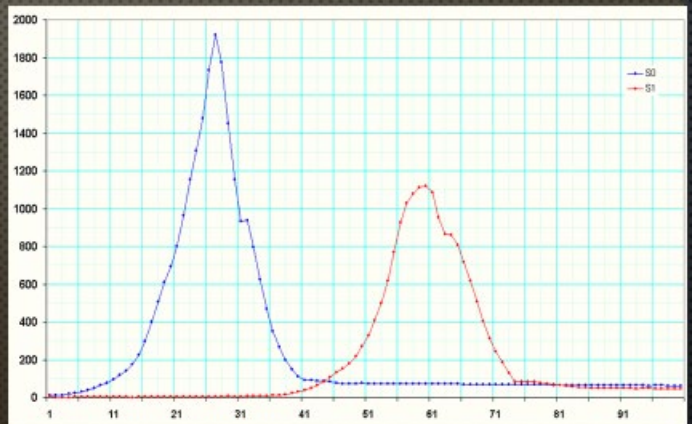


Рисунок Б 11. Слайд 11 Експериментальна частина дослідження

ВІЗУАЛІЗАЦІЯ ВИМІРЮВАНЬ

Для візуалізації отриманих в ході експериментальної перевірки результатів вимірювання збурень магнітного поля було вирішено обрати графічне середовище для розробки — LabVIEW.



Рисунок Б 12. Слайд 12 Візуалізація вимірювань

АПРОБАЦІЯ

1. «ОПТИМИЗАЦИЯ ДОРОЖНОГО ТРАФИКА С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ IOT» В ЖУРНАЛІ «МОЛОДИЙ ВЧЕНИЙ» №5 (81) ТРАВЕНЬ 2020 Р.
2. «ROAD TRAFFIC OPTIMIZATION BY IOT» НА КОНФЕРЕНЦІЇ «INFORMATION TECHNOLOGY & INTERACTIONS (SATELLITE)» В 2020 РОЦІ;
3. «ОПТИМИЗАЦІЯ ТРАФІКУ МІСТ З ВИКОРИСТАННЯМ ЗАСОБІВ IOT» VI МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА ВЗАСМОДІЇ» В 2020 РОЦІ;
4. «ОПТИМИЗАЦІЯ РУХУ АВТОТРАНСПОРТУ З ВИКОРИСТАННЯМ РОЗПОДІЛЕНОЇ СИСТЕМИ ЗБОРУ ДАНИХ НА БАЗІ ДАТЧИКІВ ХОЛЛА» НА СІМНАДЦЯТІЙ МІЖНАРОДНІЙ НАУКОВО-ТЕХНІЧНІЙ КОНФЕРЕНЦІЇ «ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ» В 2021 РОЦІ.

Рисунок Б 13. Слайд 13 Апробація

ВИСНОВКИ

1. Проаналізовано існуючі системи контролю дорожнього руху та проведено аналіз методів моніторингу та обробки даних про рух автотранспорту.
2. Для реєстрації руху автотранспорту запропоновано технологію використовувати чутливих сенсорів магнітного поля – давачів Холла.
3. Розроблено проект мікропроцесорної системи та написане програмне забезпечення для функціонування системи.
4. Виготовлено діючий макет схеми визначення швидкості та напрямку руху.
5. Отримано експериментальні результати по збуренням магнітного поля в зоні контролю давачів.
6. Запропоновано алгоритм оптимізації дорожнього трафіку.
7. Отримані результати свідчать про виконання кваліфікаційної роботи магістра у повному обсязі. Досягнуто усіх поставлених цілей роботи.

Рисунок Б 14. Слайд 14 Висновки

ДЯКУЮ ЗА УВАГУ!

Рисунок Б 15. Слайд 15 Дякую за увагу!