

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КОНОТОПСЬКИЙ ІНСТИТУТ

Кафедра електронних  
приладів і автоматики

Кваліфікаційна робота бакалавра  
**Управління електронною системою за аналізом дорожньої ситуації**

Студента гр. ЕІ-61

О.В. Сергєєнко

Науковий керівник,  
викладач

М.В. Бібик

Нормоконтроль,  
ст. викладач, к.т.н.

О.Д. Динник

## РЕФЕРАТ

Об'єктом дослідження кваліфікаційної роботи є електронні системи, які використовуються для управління транспортними потоками.

Мета роботи полягає у проектуванні апаратно-програмного рішення, яке дозволить керування світлофорами в залежності від наявності транспортних засобів на дорозі.

При виконанні роботи використовувався програмний пакет «Fritzing».

Результатом проведеної роботи є спроектоване апаратно-програмне рішення, яке дозволить керування світлофорами в залежності від наявності транспортних засобів на дорозі.

Робота викладена на 31 сторінках, у тому числі включає 4 рисунків,     таблиць, список цитованої літератури із 14 джерел.

**КЛЮЧОВІ СЛОВА:** АНАЛІЗ ДОРОЖНЬОЇ СИТУАЦІЇ, КОНТРОЛЕР СВІТЛОФОРА, ARDUINO, ПРОЕКТУВАННЯ, ЕЛЕКТРОННА СИСТЕМА.

**ЗМІСТ**

	<b>с.</b>
<b>ВСТУП</b> .....	4
<b>РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ</b> .....	5
1.1 Системи автоматизованого моніторингу транспортного потоку .....	
1.2 Інтелектуальний режим керування рухом .....	8
1.3 Адаптивні системи керування світлофорами .....	11
<b>РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ</b> .....	14
2.1 Мета та задачі .....	14
2.2 Вибір засобів реалізації.....	14
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ</b> .....	17
3.1 Проектування апаратної частини .....	17
3.2 Створення програмного забезпечення для роботи з даним приладом .....	19
<b>ВИСНОВКИ</b> .....	29
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	30

## ВСТУП

Збільшення кількості транспортних засобів на дорогах створило певну проблему як для водіїв так і для пішоходів, що заключається в нерівномірному регулюванню перехресть, або пішохідних переходів у момент низької завантаженості. Завантаження вулично-дорожньої мережі знижує ефективність використання транспорту (збільшення часу затримок, кількості зупинок, витрати палива, зносу), підвищує кількість дорожньо-транспортних пригод, збільшує знос дорожнього покриття і значно погіршує екологічну обстановку (забруднення повітряного басейну, в тому числі канцерогенними компонентами, збільшення шуму).

Поліпшення умов руху транспорту можна досягти при здійсненні заходів містобудівного характеру: будівництво мостів, тунелів, та нових магістралей. Здійснення таких проектів вимагає значних фінансових вкладень і витрат часу. Однак, при вдосконаленні управління транспортними потоками можна значно пом'якшити ситуацію. Існуючі системи автоматичного регулювання забезпечують організацію безпеки дорожнього руху на дорогах, але не враховують потік транспорту в певні часи доби, що є необхідним для розвантаження міських доріг.

Мета роботи – проектування апаратно-програмного рішення, яке дозволить керування світлофорами в залежності від наявності транспортних засобів на дорозі.

## РОЗДІЛ 1

### 1.1 Системи автоматизованого моніторингу транспортного потоку

Автоматизовані системи керування дорожнім рухом (АСКДР) - це взаємопов'язаний комплекс технічних, програмних і організаційних заходів, які збирають і обробляють інформацію про дані транспортних потоків і на основі цього оптимізують управління рухом. Завданням автоматизованих систем управління дорожнім рухом (АСКДР) є забезпечення організації безпеки дорожнього руху на дорогах.[1]

АСКДР поділяються на кілька видів:

Магістральні автоматизовані системи керування дорожнім рухом (АСКДР) координованого управління - безцентрові, централізовані і централізовані інтелектуальні.

- безцентрові АСКДР - немає необхідності створення керуючого пункту. Існує 2 модифікації безцентрових АСКДР. В одній з них роботу синхронізує головний контролер, до якого йде зв'язок від інших контролерів (лінія одна для всіх). У наступній модифікації безцентрових АСКДР від всіх контролерів йде своя лінія зв'язку.

- централізовані АСКДР - мають центр управління, зі зв'язаними з ним контролерами їх власними лініями зв'язку.

- централізовані інтелектуальні АСКДР - вони оснащені визначниками транспорту, і в залежності від завантаженості потоку можуть змінювати плани координації руху.

Загальноміські АСКДР - спрощені, інтелектуальні, з управлінням на міських дорогах з безперервним і реверсивним рухом.

- інтелектуальні АСКДР - містять потужні керуючі обчислювальні комплекси і мережу інформаційних дисплеїв, які змінюються. Ці АСКДР можуть проводити безперервний контроль потоку транспорту і можуть

керувати автоматичним адаптивним управлінням ДР, дозволяючи перерозподілити транспортні потоки по мережі.

АСКДР, як частина ІТС, виконує керуючі та інформаційні функції, основними з яких є:

- керування транспортними потоками;
- забезпечення транспортної інформацією;
- організація електронних платежів;
- управління безпекою та управління в особливих ситуаціях.

«Розглянемо класифікацію АСКДР, прийняту в США, що відносить кожен АСКДР до одного з чотирьох поколінь.

Покоління 1 – управляють рухом на окремих перехрестях. Ця система, в залежності від завантаженості доріг, може управляти світлофорами по жорсткій часовій програмі, по місцевому гнучкому регулюванню за алгоритмом пошуку розривів у транспортному потоці або ручному управлінню.

Покоління 2 – управляють рухом на ділянках дорожньої мережі (магістралях, невеликих районах) на підставі раніше розрахованих планів координації, що здійснюють синхронізацію світлофорних сигналів. На цьому рівні може бути реалізований контур диспетчерського управління.

Покоління 3 – управляють рухом в крупних районах зі складними дорожніми умовами на підставі гнучкої зміни режимів управління, їх загальної і місцевої корекції, в залежності від поточної транспортної ситуації. Розрахунок параметрів, що управляють, і введення їх в АСКДР автоматизовані. Управління за прогнозом динаміки ТП.

Покоління 4 – управління в реальному часі. Сучасний стан технології управління ДР можна характеризувати двояко. З одного боку, в області практичної автоматизації управління дорожнім рухом Україна відстала від розвинених країн не менше, ніж на 20 років: АСКДР, встановлені в більшості міст, відносяться до 1-го покоління і функціонують без систематичної технологічної підтримки.[2] Розроблені периферійні технічні засоби, які

застосовуються останнім часом, орієнтовані на роботу у складі АСКДР 1-го і 2-го поколінь. Виняток становить підсистема протизаторового управління за допомогою керованих знаків, але слід зазначити, що вибір дислокації таких знаків вимагає попередньої оцінки на моделі перерозподілу транспортних потоків. З іншого боку, відставання України в даній області не представляється нездоланим. Це обумовлено тим, що на рівні алгоритмічного забезпечення сучасна технологія управління ТП цілком розроблена і багато в чому програмно реалізована, принаймні в частині, що стосується систем 2-го і 3-го поколінь. В основному ясні і принципи управління, що реалізуються в системах 4-го покоління, але в цьому випадку основним стримуючим чинником є відсутність сучасних технічних засобів, що дозволяють реалізувати на практиці управління ТП в реальному часі.»[2]

На сьогоднішній день актуальність систем автоматизованого управління дорожнім рухом безперервно зростає. Кількість автомобілів на дорогах збільшується, а з ним - кількість аварій, травм і смертей. Уряди багатьох країн стурбовані проблемами регулювання, контролю і моніторингу дорожнього руху. «У зв'язку з глобальним переходом корпорацій і користувачів в хмари надзвичайно актуальною і ринково привабливою стає проблема захисту інформації та компонентів кіберпростору від несанкціонованого доступу, деструктивних проникнень, вірусів. Виникає необхідність створення надійної і захищеною від несанкціонованих проникнень інфраструктури кіберпростору і його компонентів (віртуальні персональні комп'ютери і корпоративні мережі) за аналогією з існуючими сьогодні рішеннями в реальному кібернетичному світі.»[2] Таким чином, кожен сервіс, який розробляється в реальному світі, повинен бути поміщений в відповідну клітинку хмари, яка об'єднує близькі по функціональності і корисні людині компоненти. У США, Канаді та Японії вже реалізовані масштабні проекти, покликані вирішити наявні проблеми. Це спеціалізовані системи автоматизованого моніторингу дорожнього руху OnStar, NEXCO Central, ECall Japan та ECall Europe.[3]

## 1.2 Інтелектуальний режим керування рухом

Інтелектуальне хмарне управління дорожнім рухом (Intelligence Cloud Traffic Control - iCTC) має на меті перенесення світлофорів в хмари, що радикально змінює всю інфраструктуру дорожнього руху на Землі і створює потенційні можливості для економії тисяч тон металу для виготовлення світлофорів, сотень тисяч кіловат електроенергії на підтримку їх працездатності, мільйонів доларів на їх установку і експлуатаційні витрати, а також зменшення часу установки і актуалізації світлофорів у віртуальній інфраструктурі міст до декількох хвилин.

Формальна модель кіберсистеми представлена у вигляді двох хмарних компонентів або механізмів: 1)  $f$  - моніторинг і управління; 2)  $g$  - виконавчі інфраструктурні механізми, які пов'язані між собою сигналами моніторингу, управління і ініціювання обох компонентів для реалізації сервісів. У моделі фігурують  $A = (f, g, \mu, v, X, R, Y, P)$  відповідно: блоки управління і виконання, сигнали моніторингу та управління, входи керуючих завдань і виконавчих ресурсів, виходи індикації стану алгоритму реалізації завдання і надання сервісу. Тут також присутні сигнали зовнішнього управління дорожнім рухом  $X = (v, p, s)$  для регулювання проїзду урядових персон, поліцейських машин і автомобілів спеціального призначення відповідно.[4]

Мобільний гаджет транспортного засобу  $G$  є головним керуючим впливом для iCloud Traffic Control, так само як і він є основним споживачем світлофорних сигналів  $L$  управління рухом автомобіля, що виводяться на лобове скло  $L = \{L_i, L_t, L_h, L_x\} = F(L, G, V, T, D, P)$ , Де  $V$  – сигнали спецуправління,  $T$  - програмований цикл автономного управління світлофором,  $D$  – накопичені інтелектуальні статистичні дані по світлофору (проспекту, району), в тому числі враховують пори року і доби,  $P$  - актуальні замовлення на маршрути руху транспорту. Створення віртуальної системи світлофорів дає можливість практично без фінансових, часових, матеріальних і енергетичних витрат розмішувати шляхом програмування нові світлофори в



віртуальному просторі, так само як і видаляти їх з хмари в процесі модернізації інфраструктури. Візуалізація на лобовому склі (мобільному моніторі) сигналів світлофора і голосове дублювання підвищить якість і безпеку дорожнього руху, знизить аварійну обстановку як для водія, так і для інфраструктури міст в цілому. Хмарний світлофор, як цифровий сигнал, на відміну від аналогового сприйняття водієм реального світлофора, є більш надійним засобом управління транспортом, в тому числі і для подальшого впровадження в дорожній рух автопілота, сприймає тільки детерміновані сигнали управління.

Учасники дорожнього руху ідентифікуються в хмарі гаджетом або iPhone, який підключається при посадці в автомобіль. Статус учасника підвищується при проходженні світлофорів, якщо маршрут руху замовляється заздалегідь. Інші учасники дорожнього руху (пішоходи, мотоциклісти і велосипедисти) також мають право замовляти маршрут, підвищуючи свій статус для використання світлофорів.[5]

Переваги хмарних сервісів іСТС-системи:

- 1) Для планети - збереження екології за рахунок зменшення забруднення навколишнього середовища, підвищення тривалості та якості життя людини, економія паливно-енергетичних ресурсів шляхом скорочення часу руху завдяки вибору оптимального маршруту, зменшення кількості і складності пробок за рахунок впровадження в інфраструктуру інтелектуальних світлофорів.
- 2) Для державних структур - поліція, дорожня інспекція - точна ідентифікація автомобілів, моніторинг позиціонування транспортних засобів в часі і просторі, включаючи викрадення, колізії, несанкціоновані маршрути. Істотне зниження аварійності завдяки прорахунку рівня безпеки маневрів, зменшення наслідків дорожньо-транспортних пригод, підвищення безпеки і комфорту учасників дорожнього руху.
- 3) Для транспортних компаній - моніторинг позиціонування і пересування транспортних засобів, квазіоптимальне виконання замовлень з перевезення

пасажирів і вантажів з точки зору мінімізації матеріальних і / або тимчасових витрат.

4) Для водія - надання сервісів, пов'язаних з прокладанням квазіоптимальних маршрутів і графіка руху з урахуванням негативних чинників існуючої інфраструктури з метою мінімізації матеріальних і тимчасових витрат в режимі

реального часу. Зниження аварійності за рахунок моніторингу закритих для візуального перегляду ділянок дороги і прорахунку рівня безпеки маневрів.

5) Для пасажирів - надання сервісів по моніторингу позиціонування і руху пасажирських транспортних засобів на зупиночних або транспортних терміналах за допомогою використання стаціонарних моніторів або мобільних гаджетів для зв'язку з відповідними хмарними сервісами. Візуалізація на екрані в автомобілі критичних точок маршруту руху транспортного засобу в реальному масштабі часу шляхом використання камер відеоспостереження.[6]

Аргументи проти впровадження хмари в масштабах країни:

1. «Порушення права на недоторканність приватного життя, оскільки теоретично хмара здійснює тотальний моніторинг всіх транспортних засобів». Насправді сьогодні існує система законного перехоплення телекомунікацій, реалізована відповідно з міжнародними вимогами. Але можливість перехоплення телефонних переговорів будь-якого абонента використовується тільки в ході слідства і з санкції суду. Є можливість відстежувати місце знаходження абонента спеціальними службами. Даний факт для законослухняних громадян ніякої проблеми не створює.

2. «Додаткові витрати на придбання апаратно-програмних мобільних гаджетів ідентифікації і орендна плата, близько 100 доларів в рік, за використання сервісів іСТС-системи». Вже сьогодні практично всі жителі планети мають такі пристрої, а економічні переваги хмари, пов'язані з економією палива і зменшенням часу поїздки, цілком гідно компенсують витрати на придбання сервісів.

Практичним прикладом впровадження компонентів іСТС-системи є дистанційний модуль "SHERLOCK" призначений для побудови розподілених систем моніторингу та управління об'єктами, включаючи мобільні. Являє собою електронний виріб, побудований на основі використання трьох новітніх технологій Mobile-to-Mobile, GPS і GPRS.

Завдання, які вирішуються за допомогою модуля:

- 1) Автоматичне визначення місцезнаходження транспортних засобів.
- 2) Управління автотранспортним парком, логістика.
- 3) Автоматизація служб таксі.
- 4) Моніторинг маршруту і розкладу руху транспорту.
- 5) Моніторинг режимів експлуатації транспортних засобів.

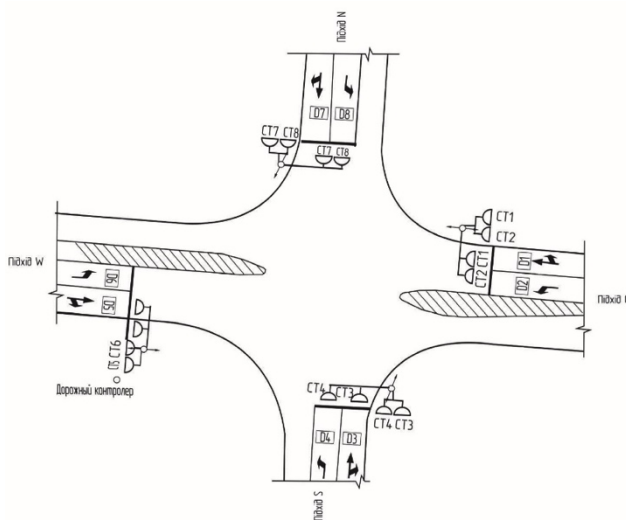
Технічні характеристики: GPS – багатоканальний приймач з високою чутливістю і малою енергоспоживанням, спеціально призначений для роботи в умовах міської забудови та наявності відображених сигналів. Доступ до онлайн-сервісу моніторингу здійснюється цілодобово із сторінки <http://gps.rfid.com.ua>. Для отримання доступу до сервісу користувач авторизується за логіном та паролем.[7]

### **1.3 Адаптивні системи керування світлофорами**

Автомобілі в міському транспортному потоці можуть відчувати значні труднощі в дорозі через неефективну роботу системи світлофорного регулювання. На етапі проектування або реконструкції дорожньої мережі дуже важливим стає вибір схеми регулювання транспортними потоками саме на перехресті, так як перехрестя є вузьким місцем вулично-дорожньої мережі (ВДМ). При цьому необхідно враховувати дорожні знаки, розмітку і параметри світлофорів. Так як кількість учасників дорожнього руху весь час збільшується, а ресурси інфраструктури вулично-дорожньої мережі обмежені то створення інтелектуальної транспортної системи стає перспективним напрямком.

Введемо поняття для позначення неадаптивної і адаптивної систем управління світлофорами. Неадаптивні системи перемикають стан світлофорів через заздалегідь заданий фіксований час. Адаптивні системи перемикають стан світлофорів з урахуванням поточної ситуації на перехресті. Використання адаптивних алгоритмів світлофорного регулювання може запобігти затори і зменшити час проїзду ділянки дороги. Але в той же час оптимізація світлофорів видається важким завданням, зважаючи на те, що необхідно управляти відразу декількома перехрестями, інакше така система може не принести ніякого ефекту.[8]

Автоматизована система керування дорожнім рухом (АСКДР) використовує алгоритми на основі даних, одержуваних з безпосередньої взаємодії з автомобілем (gps-датчики) або з встановлених на дорозі детекторів (індукційні петлі, відеокамери, інфрачервоні датчики). Детектори в реальному часі збирають деяку інформацію про транспортний потік: його інтенсивність, середню швидкість, заповненість дороги. Всі ці параметри використовуються при здійсненні прогнозу черг транспортних засобів, їх затримки і зупинки з метою оптимізації руху. Наприклад, якщо стане можливим знати, що деякі дороги стануть перевантаженими через деякий час, то ця інформація може бути передана для учасників дорожнього руху, які можуть обійти цю дорогу, тим самим вся система звільниться від заторів.



Умовні позначення:  
 СТ – світлофор транспортний;  
 D – детектор транспорту;  
 Підхід W, O, S, N – відповідно західний, східний, південний, північний підходи.

Рис. 1.1 Схема розміщення дорожніх детекторів і світлофорів для організації динамічного адаптивного регулювання руху на перехресті

Основні параметри, необхідні для організації світлофорного регулювання, такі:

- тривалість циклу ( $T_c$ );
- послідовність переключення і тривалість фаз (TF);
- тривалість зеленого сигналу світлофора ( $t_z$ );
- тривалість перехідного інтервалу ( $t_{пер}$ );
- кількість фаз у циклі (NF).

У більшості досліджень в якості вхідних змінних адаптивних алгоритмів використовуються поточна фаза світлофора, а також довжини черг на кожному з напрямків, які збираються з різних детекторів.[9]

У методі цільового пошуку керуючих параметрів в якості критерію ефективності управління використовується значення транспортної затримки на перехресті, функції якої необхідно мінімізувати.

У методі статичної оптимізації циклів світлофорного регулювання, де перехрестя являє собою систему СМО, рішення зводиться до задачі лінійного програмування з певними обмеженнями.

При включеному адаптивному режимі світлофор перемикає свій стан на мить, або при відсутності автомобілів на певній відстані у всіх відкритих напрямках, або після закінчення заданого заздалегідь часу горіння сигналу (як в неадаптивній системі). При цьому виходить наступна ситуація: якщо перед перехрестям немає черг в конфліктних напрямках, то зелений продовжує горіти для напрямлення, де автомобілів більше. Як тільки в одному з напрямків в зоні видимості з'являється автомобіль, світлофор переходить в режим неадаптивної системи, і зелений для цього конфліктного напрямку включається на заздалегідь певний, фіксований час. Такий алгоритм реалізований в програмі першого рівня системи аналітико-імітаційного моделювання транспортних мереж і потоків (САІМ ТМП).[10]

Регульоване перехрестя є найбільш важливим елементом вулично-дорожньої мережі, тому зараз у світі розроблено велику кількість

комп'ютерних програм для аналізу і моделювання як існуючих, так проєктованих регульованих перетинів (Passer V, SymTrafic, SDR, Transit-7F, Oscady, Світлофор). Це ПЗ виконує в основному схожі між собою функції (оцінка існуючої світлофорної сигналізації, розрахунок рівня завантаження, затримок і рівня обслуговування). Але, по-перше, перед використанням зарубіжних програм потрібно переконатися, що моделювання налаштоване під українські правила дорожнього руху, а, по-друге, всі перераховані вище програми мають суттєвий недолік: вони дозволяють моделювати тільки окремі перехрестя. Але реальна ситуація така, що необхідно розглядати перехрестя як елементи ВДМ всього міста. І, по-третє, не включені в стандартні ліцензії (а можуть бути реалізовані тільки через кодування на власних мовах програмування) функції, властиві сучасним інтелектуальним транспортним системам - це моделювання «зеленої хвилі», використання адаптивних алгоритмів для світлофорного регулювання.

Створення нових алгоритмів адаптивного управління і особливо впровадження їх в існуючі АСКДР дозволить ефективніше управляти транспортними потоками, контролювати та інформувати в реальному часі про зміни особистий і громадський транспорт у великих мегаполісах.[11]

## РОЗДІЛ 2

### ПОСТАНОВКА ЗАДАЧІ

#### 2.1 Мета та задачі

Метою даної роботи є проектування апаратно-програмного рішення, яке дозволить керування світлофорами в залежності від наявності транспортних засобів на дорозі.

Завдяки використанню контролера світлофора можна значно поліпшити умови руху транспорту, враховуючи щільність потоку автомобілів на дорогах.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- 1) Аналіз існуючих аналогів та вибір способу реалізації;
- 2) Проектування апаратної частини;
- 3) Створення програмного забезпечення для роботи з даним приладом.

#### 2.2 Вибір засобів реалізації

Для реалізації проекту було обрано апаратну обчислювальну платформу Arduino, яка складається із плати «мікроконтролера з елементами вводу/виводу та має середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++.

Arduino - це невелика плата з власним процесором і пам'яттю. На платі також є пара десятків контактів, до яких можна підключати всілякі компоненти: лампочки, датчики, мотори, чайники, роутери, магнітні дверні замки і взагалі все, що працює від електрики.»[11] Програмне забезпечення запускається на персональному комп'ютері і дозволяє записувати розроблений код на плату.

Основні переваги Arduino полягають в тому, що на відміну від більшості попередників, ця платформа не вимагає додаткового обладнання

(програмактора) для завантаження коду на плату - використовується простий USB-кабель. [12-13]

Плати Arduino бувають декількох видів: Nano, Mini, Uno, Leonardo, Mega, Lilyrad. Характерною рисою всіх їх є наявність програмної і апаратної частин, які призначені для певних цілей. Програмна призначена для формування алгоритмів і їх компіляції, а апаратна - це набір друкованих плат, змонтованих між собою, які продаються офіційними виробниками, а також третьою стороною.

Між різними платами є відмінності за:

- розмірами;
- кількістю виводів;
- обсягом пам'яті.

Розмір плати впливає на зручність її експлуатації. Найбільш поширеною моделлю є плата Uno, яка має розміри 6 \* 5 см. За рахунок цього можна обійтися без лупи під час створення того чи іншого прототипу на платі, а також отримати компактний пристрій, який буде добре працювати. Ця плата має малі габарити, по краях якої є 14 цифрових і 6 аналогових входів. Всі виходи характеризуються однаковими параметрами. [14]



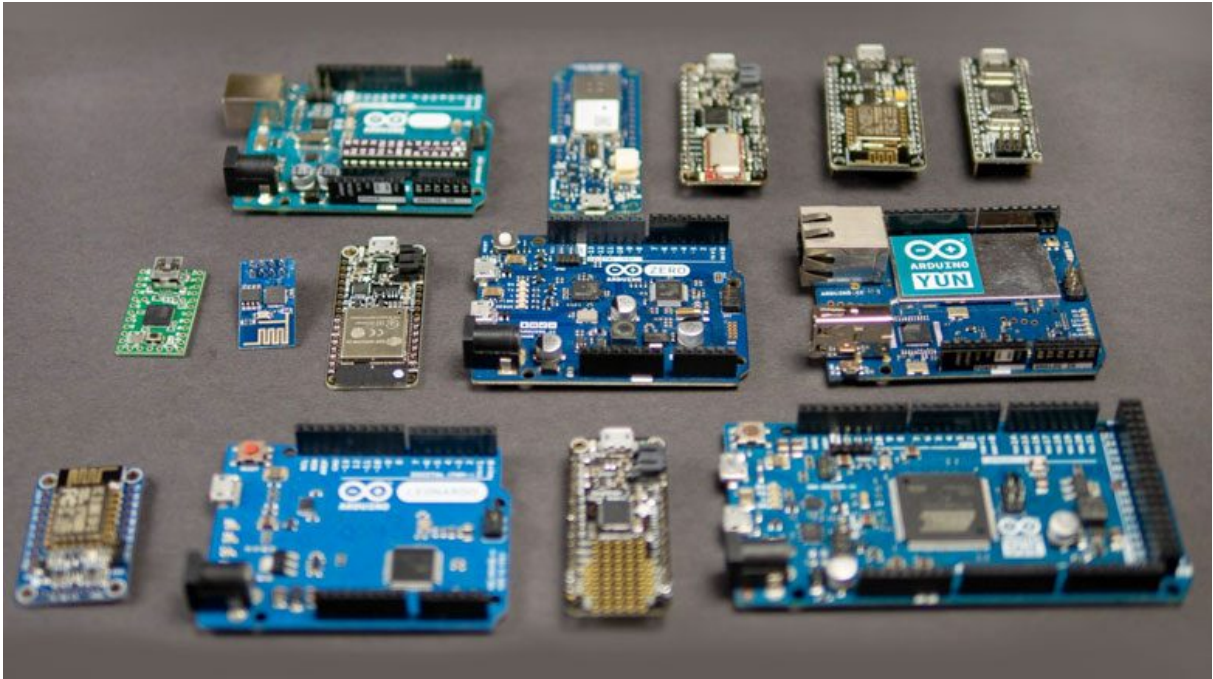


Рис. 2.1 - Зовнішній вигляд різних плат Arduino

Інші різновиди мають наступні характеристики:

- Arduino Nano - це компактна модель плати, яка має розмір 4,4 \* 1,9 см, а її функціонал не відрізняється від плат більшого розміру. Для кріплення на макетну плату використовується вихід "тато".
- Arduino Micro - це рішення виробників, яке характеризується ще меншими розмірами, а Arduino Mini - це плата з мінімальною комплектацією.
- Arduino Leonardo - це нова плата, випущена в якості ідеологічного продовження першої моделі. У неї є несуттєві відмінності в призначенні певних висновків, тому її можна швидше підключити до комп'ютера. Багато фахівців вважають за краще працювати саме з нею.
- Arduino Mega - це найбільш великогабаритна плата з усього сімейства. Її характеризує великий обсяг пам'яті і 70 виходів. Плата орієнтована на складні пристрої, які призначені для підключення різноманітної периферії.

Для проектування приладу буде використовуватися зручне середовище розробки – Fritzing. Ця програма є чудовим інструментом розробника з відкритим вихідним кодом для навчання, прототипування і обміном проектами на базі Arduino. Працює на Windows, Mac OS і Linux.

Fritzing дозволяє розробити принципову схему пристрою, і створити її подання у вигляді з'єднання макетів елементів, які виглядають навіть дуже професійно. Він також дає можливість розробити друковану плату для її подальшого виготовлення. На відміну від інших систем проектування, у Fritzing простий інтерфейс, який робить розробку електронних схем інтуїтивно зрозумілою.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОЕКТУ

#### 3.1 Проектування апаратної частини

Arduino є основною частиною цього проекту, і він буде використовуватися для зчитування сигналу з ультразвукового датчика HC-SR04 і розрахунку відстані. Ця відстань скаже нам, чи знаходиться якийсь транспортний засіб поруч з датчиком чи ні, і відповідно до цих даних буде регулюватися потік автомобілів.

Робота контролера ділиться на три етапи:

- Якщо на всіх напрямках присутній трафік, то система буде працювати нормально, перемикаючи сигнали один за одним.
- Якщо з якоїсь сторони немає трафіку, система пропустить цей сигнал і перейде до наступного. Наприклад, якщо датчик 2, 3 не реєструє наявність авто, система пропускає транспортні засоби за сигналом 1. Потім після сигналу 1 система перейде до сигналу 4, пропускаючи сигнали 2 і 3.
- Якщо на всіх 4 сигналах відсутній трафік, система зупиниться на поточному сигналі і перейде до наступного сигналу, тільки якщо на будь-якому іншому сигналі буде трафік.

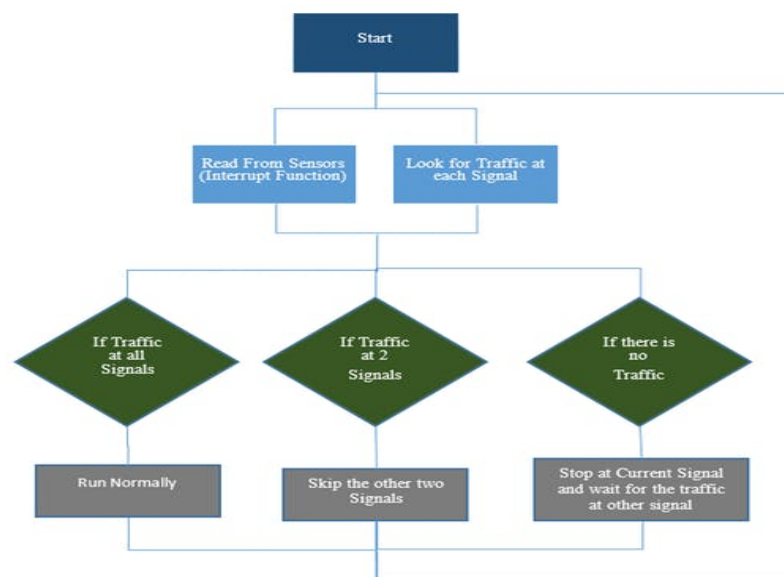


Рис. 3.1 - Функціональна схема роботи контролера світлофора

Для даного проекту необхідні наступні компоненти:

- Arduino Mega 2560
- 4 x HC-SR04 - ультразвукові датчики
- 4 червоних світлодіода
- 4 зелених світлодіода
- 4 жовтих світлодіода
- Резистори 12 x 220 Ом
- сполучні кабелі
- макетна плата

Чотири ультразвукових датчика з'єднані з Arduino. Arduino зчитує дані з цих датчиків і розраховує відстань. Цей датчик може вимірювати відстань від 2 до 400 см.

Ультразвуковий датчик випромінює ультразвукову хвилю і відбите від об'єкта відлуння приймається датчиком. Щоб генерувати хвилю, нам потрібно буде встановити тригер на 10 мкс, який відправить звуковий імпульс з 8 циклами на частоті 40 кГц, який надсилається в сторону об'єкта, і після відображення сигналу виникає луна. Потім відлуння скаже нам час, коли хвиля прийшла назад до датчика (мікросекунди). Після цього перетворимо цей час в пройдену відстань, використовуючи формулу  $S = V * T$ .

Світлодіоди підключені до Arduino через резистори 220 Ом. Необхідно використовувати резистор з світлодіодом. Резистор обмежує струм, що протікає через світлодіод. Якщо не використовувати його, світлодіод згорить. Можна використовувати резистор від 100 Ом до 10 кОм зі світлодіодом.

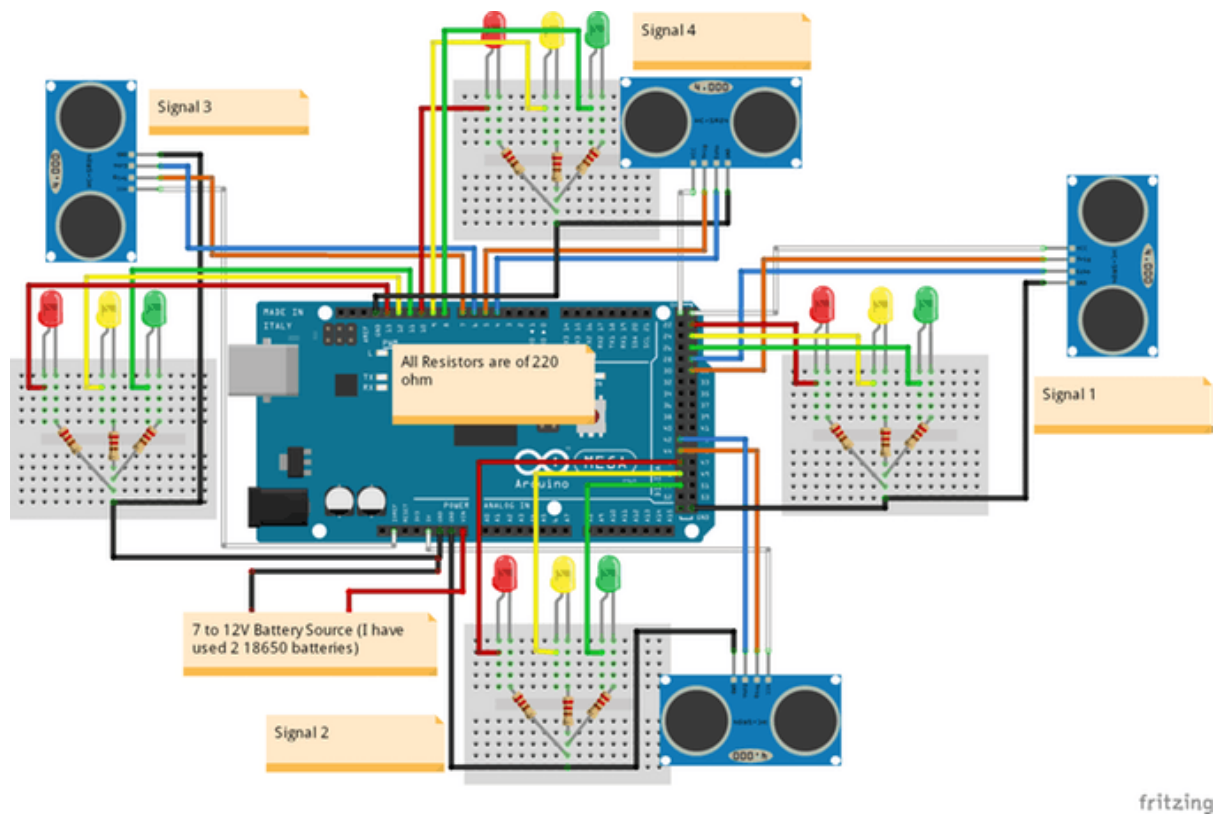


Рис. 3.2 - Схема керування світлофором

### 3.2 Створення програмного забезпечення для роботи з даним приладом

Перш за все, було включено бібліотеку `timerone`. Ця бібліотека використовується для повторного вимірювання періоду часу в мікросекундах, і в кінці кожного періоду буде викликатися функція переривання.

Бібліотека дозволить викликати функцію, в якій ми будемо безперервно зчитувати дані з датчиків, а в функції циклу будемо управляти сигналами світлофора.

```
#include <TimerOne.h>
```

```
1
```

```
#include <TimerOne.h>
```

У функції `setup` я використовував функцію `Timer1.initialize` (мікросекунди). Ця функція повинна бути викликана, перш ніж

використовувати будь-який з інших методів бібліотеки `timerone`. «Мікросекунди» - це період часу, який займає таймер. Тут можна вказати період таймера. Період за замовчуванням становить 1 секунду.

```
Timer1.initialize (100000);
```

```
1
```

```
Timer1.initialize (100000);
```

`Timer1.attachInterrupt (softInterr)` викликає функцію кожен раз, коли закінчується період таймера. Я встановив період таймера на 100000, тому наша функція буде викликатися через 100 мілісекунд.

```
Timer1.attachInterrupt (softInterr);
```

```
1
```

```
Timer1.attachInterrupt (softInterr);
```

У функції циклу перевіряється, чи є які-небудь транспортні засоби на відстанні 5 см чи ні. Якщо буде автомобіль, то буде викликана функція для включення дозволяючого сигналу в даному напрямку.

```
«void loop()
```

```
{
```

```
  // If there are vehicles at signal 1
```

```
  if(S1<t)
```

```
  {
```

```
    signal1Function();
```

```
  }
```

```
  // If there are vehicles at signal 2
```

```
  if(S2<t)
```

```
{
```

```

    signal2Function();
}
// If there are vehicles at signal 3
if(S3<t)
{
    signal3Function();
}
// If there are vehicles at signal 4
if(S4<t)
{
    signal4Function();
}
}»[14]

```

Softinterr () - це функція переривання, яка буде викликатися через кожні 100 мілісекунд. У цій функції була розрахована відстань з ультразвукових датчиків.

```

void softInterr()
{
    // Reading from first ultrasonic sensor
    digitalWrite(triggerpin1, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerpin1, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerpin1, LOW);
    time = pulseIn(echopin1, HIGH);
    S1= time*0.034/2;
}

```

Код Arduino для контролера світлофора на основі щільності трафіку виглядає наступним чином:

```

«#include<TimerOne.h>
int signal1[] = {23, 25, 27};
int signal2[] = {46, 48, 50};
int signal3[] = {13, 12, 11};
int signal4[] = {10, 9, 8};
int redDelay = 5000;
int yellowDelay = 2000;
volatile int triggerpin1 = 31;
volatile int echopin1 = 29;
volatile int triggerpin2 = 44;
volatile int echopin2 = 42;
volatile int triggerpin3 = 7;
volatile int echopin3 = 6;
volatile int triggerpin4 = 5;
volatile int echopin4 = 4;
volatile long time;
volatile int S1, S2, S3, S4;      int t = 5;
void setup(){
  Serial.begin(115200);
  Timer1.initialize(100000);  Timer1.attachInterrupt(softInterr);
  for(int i=0; i<3; i++){
    pinMode(signal1[i], OUTPUT);
    pinMode(signal2[i], OUTPUT);
    pinMode(signal3[i], OUTPUT);
    pinMode(signal4[i], OUTPUT);
  }»[14]

  pinMode(triggerpin1, OUTPUT);
  pinMode(echopin1, INPUT);

```



```
pinMode(triggerpin2, OUTPUT);
pinMode(echopin2, INPUT);
pinMode(triggerpin3, OUTPUT);
pinMode(echopin3, INPUT);
pinMode(triggerpin4, OUTPUT);
pinMode(echopin4, INPUT);
}
void loop()
{
  if(S1<t)
  {
    signal1Function();
  }
  if(S2<t)
  {
    signal2Function();
  }
  if(S3<t)
  {
    signal3Function();
  }
  if(S4<t)
  {
    signal4Function();
  }
}

void softInterr()
{
  digitalWrite(triggerpin1, LOW);
```

```
delayMicroseconds(2);  
digitalWrite(triggerpin1, HIGH);  
delayMicroseconds(10);  
digitalWrite(triggerpin1, LOW);  
time = pulseIn(echopin1, HIGH);  
S1= time*0.034/2;
```

```
digitalWrite(triggerpin2, LOW);  
delayMicroseconds(2);  
digitalWrite(triggerpin2, HIGH);  
delayMicroseconds(10);  
digitalWrite(triggerpin2, LOW);  
time = pulseIn(echopin2, HIGH);  
S2= time*0.034/2;
```

```
digitalWrite(triggerpin3, LOW);  
delayMicroseconds(2);  
digitalWrite(triggerpin3, HIGH);  
delayMicroseconds(10);  
digitalWrite(triggerpin3, LOW);  
time = pulseIn(echopin3, HIGH);  
S3= time*0.034/2;
```

```
digitalWrite(triggerpin4, LOW);  
delayMicroseconds(2);  
digitalWrite(triggerpin4, HIGH);  
delayMicroseconds(10);  
digitalWrite(triggerpin4, LOW);  
time = pulseIn(echopin4, HIGH);  
S4= time*0.034/2;
```

```
    Serial.print("S1: ");
    Serial.print(S1);
    Serial.print(" S2: ");
    Serial.print(S2);
    Serial.print(" S3: ");
    Serial.print(S3);
    Serial.print(" S4: ");
    Serial.println(S4);
}
void signal1Function()
{
    Serial.println("1");
    low();

    digitalWrite(signal1[0], LOW);
    digitalWrite(signal1[2], HIGH);
    delay(redDelay);
    // if there are vehicels at other signals
    if(S2<t || S3<t || S4<t)
    {

        digitalWrite(signal1[2], LOW);
        digitalWrite(signal1[1], HIGH);
        delay(yellowDelay);
    }
}
void signal2Function()
{
    Serial.println("2");
    low();
```

```
digitalWrite(signal2[0], LOW);
digitalWrite(signal2[2], HIGH);
delay(redDelay);

if(S1<t || S3<t || S4<t)
{
  digitalWrite(signal2[2], LOW);
  digitalWrite(signal2[1], HIGH);
  delay(yellowDelay);
}
}
void signal3Function()
{
  Serial.println("3");
  low();
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal3[2], HIGH);
  delay(redDelay);
  if(S1<t || S2<t || S4<t)
  {
    digitalWrite(signal3[2], LOW);
    digitalWrite(signal3[1], HIGH);
    delay(yellowDelay);
  }
}
void signal4Function()
{
  Serial.println("4");
  low();
  digitalWrite(signal4[0], LOW);
```

```
digitalWrite(signal4[2], HIGH);
delay(redDelay);
if(S1<t || S2<t || S3<t)
{
    digitalWrite(signal4[2], LOW);
    digitalWrite(signal4[1], HIGH);
    delay(yellowDelay);
}
}
void low()
{
    for(int i=1; i<3; i++)
    {
        digitalWrite(signal1[i], LOW);
        digitalWrite(signal2[i], LOW);
        digitalWrite(signal3[i], LOW);
        digitalWrite(signal4[i], LOW);
    }
    for(int i=0; i<1; i++)
    {
        digitalWrite(signal1[i], HIGH);
        digitalWrite(signal2[i], HIGH);
        digitalWrite(signal3[i], HIGH);
        digitalWrite(signal4[i], HIGH);
    }
}
```

## ВИСНОВКИ

1. В рамках цієї роботи був проведений аналіз предметної області, обрано методи та засоби реалізації. Сучасні існуючі системи автоматичного регулювання забезпечують організацію безпеки дорожнього руху на дорогах, але не враховують потік транспорту в певні часи доби, що є необхідним для розвантаження міських доріг.
2. Спроектовано апаратно-програмне рішення, яке дозволить керування світлофорами в залежності від наявності транспортних засобів на дорозі.
3. Практичне застосування даної розробки зосереджено на поліпшенні умов руху транспорту, що може значно допомогти в розвантаженні міських доріг.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бесекерский В.А. Теория систем автоматического управления / Бесекерский В.А., Попов Е.П. - СПб.: Профессия, 2003. - 747 с.
2. Кобдииков М.А. Работа в автоматизированной системе оперативного управления перевозочным процессом / Кобдииков М.А., Мустапаев А.Д. - Алматы: КазАТК, 1998 г. – 140 с.
3. Кір'янов О.Ф. Інформаційні технології на автомобільному транспорті / Кір'янов О.Ф., Мороз М.М. - м.Кременчук: Кременчуцький національний університет ім. М.Остроградського, 2013. - 300 с.
4. Хаханов В.И. «Зеленая волна» – облако мониторинга и управления дорожным движением / Хаханов В.И., Меликян В.Ш, Саатчян А.Г., Шахов Д.В. - Армения.: Вестник «Информационные технологии, электроника, радиотехника». Вып. 16(№1), 2013. - 60 с.
5. Pandit A.A., Talreja J., Mundra A.K. RFID Tracking System for Vehicles (RTSV) // First International Conference on Computational Intelligence, Communication Systems and Networks, 2009. - P.160-165.
6. Jiang Lin-ying, Wang Shuai, Zhang Heng, Tan Han-qing. Improved Design of Vehicle Management System Based on RFID // International Conference on Intelligent System Design and Engineering Application (ISDEA). 2010. - P. 844-847.
7. Schutte J. Recent trends in automatic train controls // IEEE Intelligent Transportation Systems. 2001. - P. 813 -819.
8. Протодьяконов А.В. Оценка эффективности светофорного регулирования на перекрестке при использовании адаптивной нейро-нечеткой системы управления / Протодьяконов А.В., Швец С.Е., Фомин А.Н. - 2010. - С. 211- 220.
9. Задорожный В.Н. Аналитико-имитационные методы решения актуальных задач системного анализа больших сетей / Задорожный В.Н.,

Долгушин Д. Ю., Юдин Е.Б. - Омск : изд-во ОмГТУ, 2013. – С. 86-125

10. Пугачев И.Н. Организация и безопасность движения: учебное пособие / Пугачев И.Н. – Хабаровск: Изд-во Хабар. гос. техн. ун-та, 2004. – 232с.

11. Голуб Д.И. Метод статической оптимизации ЦИКЛОВ светофорного регулирования / Голуб Д.И. - СевКавГТУ Сер. Экономика, 2007.

12. Джереми Б. Изучаем Arduino / Джереми Б. – СПб.: Издательский центр «БХВ-Петербург», 2015. – 401с.

13. Петин В. Проекты с использованием контролера Arduino / Петин В. - СПб.: Издательский центр «БХВ-Петербург», 2014. – 333с.

14. <https://doc.arduino.ua/ru/hardware/Mega2560> - Платы Ардуино Mega 2560, дата доступа: 18.04.2020 р.