

Міністерство освіти і науки України  
Чорноморський національний університет імені Петра Могили  
Факультет комп'ютерних наук  
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

В. о. завідувача кафедри,  
канд. тех. наук, доцент

\_\_\_\_\_ Я. М. Крайник

« \_\_ » \_\_\_\_\_ 2022 р.

## МАГІСТРСЬКА РОБОТА

Галузь знань: 12 Інформаційні технології  
Спеціальність: 6.050102 Комп'ютерна інженерія  
Тема: **Апаратне забезпечення та система управління для триколісного транспортного засобу**  
Шифр: 6.050102 – МР.ПЗ.00 – 605.21610611

Виконав:

студент 4 курсу, групи 406,  
спеціальності  
6.050102 Комп'ютерна інженерія  
М. К. Чупріков  
\_\_\_\_\_

Керівник:

канд. тех. наук, доцент  
Я. М. Крайник  
\_\_\_\_\_

Миколаїв 2022

Апаратне забезпечення та система управління для триколісного транспортного засобу  
**ЗАВДАННЯ**

на виконання бакалаврської роботи

**НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!**

**ЗАРЕЗЕРВОВАНА** Сторінка 1

**ця сторінка після друку буде замінена**

Апаратне забезпечення та система управління для триколісного транспортного засобу  
**ЗАВДАННЯ**

на виконання бакалаврської роботи

**НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!**

**ЗАРЕЗЕРВОВАНА Сторінка 2**

**ця сторінка після друку буде замінена**

АНОТАЦІЯ

1 сторінка !!!!

**НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!**

**ЗАРЕЗЕРВОВАНА Сторінка 1**

**ця сторінка після друку буде замінена**

ABSTRACT

1 сторінка !!!!

**НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!**

**ЗАРЕЗЕРВОВАНА Сторінка 2**

**ця сторінка після друку буде замінена**

## ЗМІСТ

|   |  |
|---|--|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....  | 8                                      |
| ВСТУП .....   | 9                                      |
| РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ІНФОРМАЦІЇ ПО ПРИНЦИПУ<br>РОБОТИ КОМПЛЕКСНИХ СИСТЕМ ЗАБЕЗПЕЧУЮЧИХ ПЕРЕСУВАННЯ<br>..... | 11                                     |
| 1.1 Сегвей.....   | 11                                     |
| 1.2 Гіроскутер .....  | 15                                     |
| 1.3 Електросамокати .....   | 17                                     |
| 1.4 ЕЛЕКТРОКАРИ .....   | 20                                     |
| 1.5 Переваги та недоліки .....  | 20                                     |
| 1.5.1 компромісне рішення.....  | <b>Ошибка! Закладка не определена.</b> |
| Висновки до розділу 1 .....   | 24                                     |
| РОЗДІЛ 2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ ПРОГРАМНО-<br>АПАРАТНОГО КОМПЛЕКСУ .....                                      | 25                                     |
| 2.1 Огляд обладнання необхідного для створення апаратного комплексу<br>.....                                      | 25                                     |
| 2.1.1 Мікрокомп'ютер Raspberry Pi 4 .....   | 25                                     |
| 2.1.2 Електро двигун .....  | 29                                     |
| 2.1.3 Регулятор – Акумулятор .....  | 30                                     |
| 2.1.4 Камера .....  | 34                                     |
| 2.1.5 Гальмівна система.....  | 34                                     |
| 2.1.6 Рама конструкції .....  | 36                                     |
| 2.2 Опис функціонування схеми.....  | 37                                     |
| 2.3 Опис функціонування окремих частин схеми. ....  | 38                                     |
| 2.3.1 Ручний режим керування .....  | 38                                     |
| 2.3.2 Режим віддаленого використання .....  | 39                                     |
| Висновки до розділу 2 .....   | 40                                     |
| Розділ 3 Розробка програмної частини Пристрою .....   | 41                                     |

|          |  |    |
|----------|--|----|
| 3.1      | Опис середовища розробки Arduino IDE.....                    | 41 |
| 3.2      | Налаштування Orange Pi .....                                 | 42 |
| 3.3      | Програмна частина керування .....                            | 45 |
| 3.4      | Опис користувацького інтерфейсу та тестування пристрою ..... | 46 |
|          | Висновки до розділу 3 .....                                  | 47 |
| Розділ 4 | ТЕСТУВАННЯ ПРИСТРОЮ, ТА ІНСТРУКЦІЯ З .....                   | 48 |
| 4.1      | Тестування пристрою .....                                    | 48 |
| 4.1.1    | Тестування двигуна та інвертора .....                        | 48 |
| 4.1.2    | Тестування апаратно-програмної частини комплексу .....       | 54 |
| 4.1.3    | Тестування акумуляторної батареї пристрою .....              | 58 |
| 4.1.4    | Тестування сервера .....                                     | 61 |
| 4.1.5    | Тестування гальмівної системи .....                          | 64 |
| 4.2      | Інструкція з використання Для використання .....             | 65 |
|          | Висновки до розділу 4 .....                                  | 65 |
|          | ВИСНОВКИ.....  | 66 |
|          | ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....                               | 67 |
|          | Додаток А  |    |
|          | Принципова схема пристрою.....                               | 68 |
|          | Додаток Б  |    |
|          | Програмний код пристрою.....                                 | 70 |

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

|      |   |   |
|------|---|---|
| AGM  | – | Absorbed Glass Mat                          |
| IDE  | – | Integrated Development Environment          |
| ISP  | – | In-system Programming                       |
| LCD  | – | Liquid Crystal Display                      |
| RISC | – | Reduced Instruction Set Computer            |
| SPI  | – | Serial Peripheral Interface                 |
| UART | – | Universal Asynchronous Receiver-Transmitter |
| USB  | – | Universal Serial Bus                        |
|      |   |   |
| АЦП  | – | Аналого-цифровий перетворювач               |
| ДБЖ  | – | Джерело безперебійного живлення             |
| МК   | – | Мікроконтролер                              |
| ОЗП  | – | Оперативний запам'ятовуючий пристрій        |
| ОП   | – | Операційний підсилювач                      |
| ПК   | – | Персональний комп'ютер                      |
| ШІМ  | – | Широтно-імпульсна модуляція                 |



## ВСТУП

Завдяки технологічному прогресу останнього тридцятиліття в області мікроелектроніки та комп'ютерної техніки, в 2022 існує велика кількість механічно електронних приладів які здатні пересуватися як за рахунок колісного приводу, наприклад автомобілі або іграшкові моделі автомобілів, так і за рахунок гвинтокрилої тяги, наприклад гелікоптер, або дрон, так і за рахунок турбо реактивної тяги, яку може забезпечити лопатева турбіна, яка може знайти застосування на літаках та іншій техніці, при комбінуванні даних властивостей можна забезпечити пересування в декількох площинах.

На сьогоднішній день активного вжитку здобули колісні сегвеї, електросамокати та інші види даного сегменту, вони можуть розвивати досить високу швидкість пересування та працювати від джерела електроживлення таких як акумулятор на протязі 4-7 годин.

Даний вид транспорту не забезпечую досить великий рівень безпеки оскільки не всі моделі мають програмне обмеження швидкості для забезпечення безпечності, а водії таких транспортів часто нехтують засобами особистого захисту такими як шолом або наплічники. Тому є необхідність створення апаратного комплексу який буде мати свій власний функціонал та зможе забезпечувати рівень безпеки.

**Мета:** розробити апаратне забезпечення та систему управління загального призначення з можливістю пересуватися з відповідними нормами безпеки з відносно великим ресурсом робочих часів використання.

**Об'єкт:** особливості використання та методи пересування об'єктів в просторі з урахуванням норм безпеки, шляхи керування та стабілізації таких моделей в просторі.

**Предмет:** пристрій апаратного забезпечення та системи керування для триколісного засобу .

Для досягнення поставленої мети необхідно вирішити такі **завдання:**

- проаналізувати методи пересування об'єктів в просторі.

Апаратне забезпечення та система управління для триколісного транспортного засобу

– проаналізувати функціональні можливості наявних пристроїв аналогічного призначення;

– спроектувати апаратну частину пристрою, використовуючи загальнодоступні компоненти, передбачивши можливість подальшої модифікації та вдосконалення;

– розробити програмну частину для забезпечення функціонування спроектованого апаратного модулю;

– виготовити макетний зразок модулю та перевірити його працездатність.

**Практичне значення** розроблений прототип дає змогу безпечного пересування, система моніторингу може забезпечувати запис відео з навколишнього середовища та відтворення його на мобільному пристрої людини яка контролює дії водія транспорту, система також може забезпечувати дистанційне керування транспортним засобом за допомогою інтерфейсу передачі даних WiFi .

## РОЗДІЛ 1

### АНАЛІТИЧНИЙ ОГЛЯД ІНФОРМАЦІЇ ПО ПРИНЦИПУ РОБОТИ КОМПЛЕКСНИХ СИСТЕМ ЗАБЕЗПЕЧУЮЧИХ ПЕРЕСУВАННЯ

В наш час відомо багато різних типів пристроїв які приводяться в рух за рахунок перетворення електричної сили в механічну, велика кількість таких комплексів використовується як в промисловій сфері використання так і в медичному або військовому, до таких комплексів можна віднести:

- гіроскутери;
- сегвей;
- електрокари
- електросамокати.

Кожному конкретному типу властиві свої переваги та недоліки, і кожен з них потребує певних умов експлуатації.

#### 1.1 Сегвей

Це електричний транспортний засіб, оснащений механізмом самобалансування. Автоматична система реагує на зміну положення корпусу людини, що стоїть на платформі. Сигнали з гіроскопічних і рідинних датчиків надходять в ядро процесора, який управляє двигунами. У мікропроцесор надходить до ста сигналів в секунду..



Рисунок 1.1 – Сегвей

Сегвей (рис. 1.1) має Два колеса розташовані співвісно. Транспортний засіб Segway автоматично балансується за зміни положення корпусу їздка; для цієї мети використовується система індикаторної стабілізації: сигнали з гіроскопічних та рідинних датчиків нахилу надходять на мікропроцесори, які виробляють електричні сигнали, що впливають на двигуни та управляють рухами. Кожне колесо транспортного засобу Segway приводиться у обертання своїм електродвигуном, що реагує на зміни рівноваги машини.

При нахилі тіла їздця вперед сегвей починає котитися вперед, при збільшенні ж кута нахилу тіла їздця швидкість сегвея збільшується. У разі відхилення корпусу назад самокат уповільнює рух, зупиняється або котиться заднім ходом. Керування в першій моделі відбувається за допомогою поворотної рукоятки, в нових моделях - хитання колонки вліво-вправо.

Транспортний засіб Segway розвиває швидкість близько 50 км/год та має власну масу близько 40 кг (без батареї), його ширина 60 см, а допустиме навантаження – 140 кг. Акумулятор забезпечує пробіг до 39 км. Залежно від моделі ці показники можуть змінюватись. Спеціально для оснащення ними пристрою транспортного засобу Segway розроблено компактні, але потужні (2 л. с.) електродвигуни. Кожен із них через редуктор пов'язаний зі своїм колесом. Самокат може рухатися не лише по асфальту, а й по ґрунту.

Щоб почати пересування, потрібно трохи схилити тіло вперед. Чим сильніший кут нахилу, тим вища швидкість. При відведенні корпусу трохи назад швидкість буде сповільнюватися, потім станеться зупинка або транспортний засіб автоматично переключиться на задній хід. Це залежить від інтенсивності та швидкості рухів водія. Повороти здійснюються за допомогою керма.

Пересування на сигвеї не потребує спеціального навчання. Це відрізняє автомобіль від іншого особистого транспорту. Електроскутер самостійно зберігає баланс. Система дуже проста та доступна. Навчитися кататися на сигвеї з нуля можна буквально за десять-п'ятнадцять хвилин.

Принцип роботи пристрою надзвичайно простий та надійний. З боків платформи знаходяться колеса, які розташовані на одній осі. Кожне з них обладнане індивідуальним електромотором, який передає тягу на колесо за допомогою редуктора, що підсилює. Не можна не помітити важіль управління сегвеєм, що піднімається над платформою, як щогла корабля. Він і органом управління є, і одночасно точкою опори для їздця. Для зручності використання розробники передбачили регулювання цього елемента за висотою. За це відповідає затискний болт: відкрутили, відрегулювали потрібну висоту, зафіксували болтом.



Рисунок 1.2 – Структура сегвея

У структуру гаджета (рис. 1.2) впроваджено сильний комп'ютер, який би прискорення чи уповільнення електромоторів. Орієнтирами для цих дій є датчики позиції наїзника і важеля управління. Система стабілізації спрацьовує миттєво, воно й не дивно, адже це безпосередньо впливає на безпеку руху.

При здійсненні поворотного маневру, колеса у сегвея не повертатимуться як у машини або велосипеда, вони функціонують за принципом уповільнення. Одне колесо зупиняється, а інше починає крутитися швидше, що і призводить до розвороту.

Апаратне забезпечення та система управління для триколісного транспортного засобу Segway можуть мати у своєму розпорядженні вхід для заряджання. Знайти його можна як на передній частині пристрою, так і позаду. Усі підряд сигвеї обладнуються світловими індикаторами зарядки, якими можна визначитися, на скільки кілометрів ще вистачить заряду акумулятора. Під ногами райдера зазвичай знаходиться спідометр.



Рисунок 1.3 – Структура схема сегвея

Роль накопичувача грає літій-іонна батарея, яка додає засобу пересування істотну частину маси. Без акумуляторної батареї вага сегвею становить 15-45 кг, залежно від моделі. Колеса та рама можуть витримати до 140 кг, проте на ринку присутні посилені модифікації, які можуть прийняти на борт райдера вагою до 200 кг.

Бюджетні моделі segway можуть запропонувати максимальну швидкість до 20 км/год, тому що саме такий швидкісний режим вважається найбезпечнішим. А ось засоби пересування, що відносяться до категорії PRO, можуть забезпечити граничну швидкість 50 км/год! Все ж таки розганятися до таких показників на сигвеї не рекомендується, оскільки його кліренс не великий і можна по необережності нарватися на якусь непереборну перешкоду.

Тривалість їзди залежить від ємності акумуляторної батареї та потужності силових агрегатів. Як правило, цифра пробігу однією заряді сягає

Апаратне забезпечення та система управління для триколісного транспортного засобу 40 км. Є на ринку і більш довгограючі моделі, однак вони важать досить багато, тому що накопичувач великої ємності важить дуже і дуже не мало.

## 1.2 Гіроскутер

Гіроскутер - це по-справжньому новий засіб пересування, який з'явився зовсім недавно. До цього у світі гіробордів панував Сігвей. Він був занадто великим і багато важив, тому він не став популярним у нас в країні. Як і будь-який засіб пересування у гіроскутера є свої плюси та мінуси.



Рисунок 1.4 – Гіроскутер

На Рис. 1.4 зображено гіроскутер

Гіроскутер має певні параметри та розміри. Але розміри залежать більше від розміру коліс та класу самого гіроскутера. Якщо гіроскутер має невеликі 4.5 дюймові колеса. Такий гіроборд маленький і підійде тільки для дітей. Є моделі з 6.5 дюймовими колесами, ця модель також добре підійде дітям і підліткам. Маленькі колеса мають один невеликий недолік, вони дуже погано їздять, по поганих дорогах.

Через малий розмір коліс, зазор між землею і нижньою частиною дошки занадто маленький і тому, по глибоких ям, ви не зможете проїхати. Також такі гіроскутери мають безкамерні колеса, тому їзда на такому гіроскутері не така плавна як на моделях з 10-ти дюймовими колесами.

Є гіроскутери із середнім 8 дюймовим розміром коліс. Цей гіроскутер має не найбільші колеса, але при цьому дуже потужні характеристики. Якщо ви не знаєте який гіроскутер взяти, то беріть цю модель. Він підійде майже всім.

10-ти дюймові моделі добре підходять для дорослих чоловіків і тих хто любить швидко їздити. Це швидкі та потужні моделі, які можуть їздити практично будь-якими поверхнями. Через великий зазор внизу, він може їздити навіть по невеликих ямах, калюжах і траві.

Розглянемо потужність акумуляторів. Практично у всіх гіроскутерів потужність однакова. У середньому вона становить 4400 мА/год. Але в чому різниця, чому ж багато гіроскутерів коштують, дорожче, а багато дешевших? Справа в тому, що всі акумулятори, при цьому різні за класом.

Є гіроскутери з низьким класом типу 1С та 2С. Такі батареї зазвичай ставлять на гіроскутери з маленькими колесами - 4.5 і 6.5 дюймів. Ці гіроскутери призначені для їзди тільки рівними дорогами, тому їм акумулятори високого класу і не потрібні.

Гіроскутери ж з 8-ми дюймовими колесами мають вже потужніші акумулятори. Найчастіше на ці моделі ставлять батареї 3С класу. Тут ще грає роль і вага самого гіроскутера. Чим більша вага та розмір коліс, тим вище клас акумулятора потрібен цій моделі.

У найбільших гіроскутерів з 10-ти дюймовими надувними колесами стоять акумулятори найвищого класу 4С та 5С. Для їзди бездоріжжям і поганим дорогам, потрібна надійна батарея, щоб ваш гіроскутер не відключався і не зменшував швидкість.

Потужність електродвигунів також залежить від розміру самого гіроскутера. Чим більше гіроскутер, тим потужніше електродвигуни йому ставлять. У дитячих моделей, зазвичай, стоять малопотужні двигуни по 200 - 250 Ватт. У гіроскутера 6.5 і 8 дюймів зазвичай ставлять електродвигуни середньої потужності по 350 Ватт на кожне колесо. Такі гіроскутери здатні



Апаратне забезпечення та система управління для триколісного транспортного засобу розганяється до швидкості 15 км/год. Правда на такій максимальній швидкості досить сильно сідає батарея.

Потрібно ще обов'язково пам'ятати, що чим вища вага водія, тим нижча швидкість і тим швидше сідає акумулятор. Якщо гіроскутер їздить на піковому навантаженні, швидкість може сильно просідати, особливо в ділянках з поганою дорогою. Добре, що середнє максимальне навантаження на платформу у всіх гіроскутерів, дорівнює показнику 120 кг. Також ця характеристика зростає зі збільшенням класу гіроскутера. Є моделі з 10 дюймовими колесами, які можуть перевозити до 135 кг.

Максимальна швидкість у самого швидкого гіроскутера, становить 25 км/год. Щоправда, щоб розігнатися до такої неймовірної швидкості, потрібна ідеально рівна дорога, без ухилу, тріщин і дорожнього малюнка, як, наприклад, на плитці.

Гіроскутер має найголовнішу особливість та перевагу перед багатьма гіробордами. Це звичайно розмір і компактність. Його легко можна засунути в сумку або багажник машини, а маленькі моделі можна просто нести в руках. Він займає мало місця вдома. І ви навіть не помітите його, якщо поставите в кут кімнати. Головним плюсом є звичайно ж ціна. Це найдешевший гіроборд із усіх.

### 1.3 Електросамокати



Рисунок 1.5 – Електросамокат

Конструкція електросамоката (рис. 1.5) схожа на звичайний, за винятком того, що в ньому з'явився електромотор та все, що необхідно для його роботи. Основними вузлами електросамокату є:

- Несуча рама. Це основа конструкції, на якій навішане та закріплене все інше обладнання. Виготовляється з металу (в основному алюміній та його сплави) або карбону;
- Рульова колонка (стійка). З її допомогою здійснюється керування самокатом за допомогою ручок та органів керування електромотором. Зручніше, якщо вона телескопічна, тому що в цьому випадку можна налаштувати висоту під конкретне зростання людини;
- Дека. Так називається платформа, яку власник самоката стає під час їзди. Дека жорстко кріпиться до несучої рами. Зверху вона зазвичай покрита шаром гуми або має рифлену поверхню, часто з якимось малюнком для забезпечення кращого зчеплення із взуттям. Внизу у деки є ніша, в якій знаходиться акумулятор і електроніка, що управляє.

Колеса та електродвигун. Зазвичай на самокаті переднє колесо є поворотним веденим, заднє – провідним. Втім, бувають і передньопривідні варіанти, а також моделі, у яких обидва колеса є провідними. Провідне колесо буває двох типів. Перший варіант – мотор-колесо. В цьому випадку електропривод вбудований безпосередньо в колесо. Другий — провідне колесо, що рухається ланцюговою передачею від встановленого на рамі електродвигуна. Також колеса бувають камерними та безкамерними та розрізняються по діаметру в діапазоні від 4 до 12 дюймів.

Гальма. В електросамокатах використовують електричне або дискове гальмо. Перший не дозволяє виконати швидку зупинку, зате під час використання трохи заряджає акумулятор і не вимагає обслуговування. Дискове гальмо ефективно гасить високу швидкість, але потребує

Апаратне забезпечення та система управління для триколісного транспортного засобу регулярного обслуговування. Крім того, є гальмо за допомогою заднього крила - при гальмуванні нога ставиться на заднє крило, і його натисканням блокується обертання заднього колеса. Гальмо заднім крилом є доповненням до електричного чи дискового.

Силова та керуюча електроніка. Це в першу чергу двигун, акумулятор і контролер, що управляє роботою електромотора в залежності від команд, що надходять (додати швидкість або гальмувати). Інформація про роботу двигуна видається на дисплей на кермі.

Амортизатори. Залежно від конструкції бувають газонаповненими чи пружинними. Ряд моделей, переважно недорогі, взагалі не оснащуються амортизаторами. Як варіант, виробниками заявляється, що амортизатором є пневматичні колеса. Такі колеса насправді амортизують нерівності у процесі руху, але повноцінною заміною амортизаторам не є.

Працює електросамокат в такий спосіб. Акумуляторна батарея приводить у дію електродвигун, конкретні параметри роботи залежать від встановленого режиму. Режимом роботи управляє контролер, який орієнтується на команди водія, що віддаються з рукоятки акселератора та інформацію з датчиків. Як датчик використовується гіроскоп, який фіксує кут нахилу до дороги. Електродвигун формує обертальний момент, що передається на колесо. Колеса обертаються, самокат рухається, ви їдете.

Сучасні електросамокати мають функцію зворотного зв'язку або рекуперації. Це означає, що коли ви пересуваєтеся не за рахунок роботи електромотора, а, наприклад, їдете під гірку або відштовхуєтеся ногою, то колеса обертаються вільно і не витрачають заряд акумулятора, а навпаки, заряджають його. Звичайно, це не дозволить повністю зарядити батарею, вічних двигунів у природі не існує. Але трохи підзарядки та збільшення дальності поїздки нікому не завадить. Тим більше, що це одна з основних характеристик електросамокату.

## 1.4 ЕЛЕКТРОКАРИ



Рисунок 1.6 – Електромобіль renault twizy

Електромобіль (рис. 1.6) – це транспортний засіб, який працює на електричній тязі. Це звичайнісінький автомобіль, але замість двигуна внутрішнього згоряння у нього встановлений електропривод, який живиться від акумуляторної батареї.

Принцип роботи електромобіля полягає в тому, що електрика від акумуляторної батареї живить електродвигун, який обертає колеса. Для контролю оборотів автомобіль оснащений педаллю газу.

Особливості цього виду транспорту такі:

Електричний двигун здатний розвивати більшу кількість оборотів, аж до 12 000 об/хв. Це означає, що автомобілю не потрібен багатоступінчастий редуктор, а відповідно коробка перемикання передач. Втім, деякі електрокар все ж оснащуються автоматичними коробками перемикання передач.

Весь момент двигуна, що крутить, доступний з нуля обертів. Це означає, що в момент натискання на педаль газу на колеса передається вся потужність двигуна.

Електричний мотор не потребує частого регулярного обслуговування (заміни масла, фільтрів і так далі), так як має мінімум частин, що труться.

Варто зазначити, що комерційні електромобілі вже давно використовуються в різних країнах. До них відносяться електровантажівки, а також складські машини, такі як навантажувачі, штабелери тощо. І їхні переваги вже підтверджені на практиці.

## 1.5 Переваги та недоліки

Розвиток в сфері перетворення енергії дав нам велику кількість шляхів за допомогою яких ми можемо створювати пристрої які мають змогу пересуватися в просторі комфортно для людини. Всі пристрої прийшли від звичайного електродвигуна до чогось більше. Розглянемо всі приведені пристрої та їх можливості під кутом зручності, надійності, безпечності.

Сігвеї можна розглядати як точку відліку оскільки даний представник портативного пристрою пересування був створений значно раніше за гіроскутери та електросамокати. Сігвей являє собою моно колесо досить великого розміру за рахунок чого акумулятор не має досить місця для розташування в пристрою що негативно впливає на ресурс використання. Сігвей вимагає від користувача тримання рівноваги в двох площинах що вимагає певних навичок в процесу його використання, моделі сігвейів на яких не наявний руль, вимагають ще більше навичок, якщо даний тип має вбудований обмежувач швидкості то цей варіант ще можна вважати досить безпечним, але якщо конструкція цього не має, або користувач навмисно його видалив, то дана конструкція взагалі може представляти потенційну загрозу як водію так і випадковим пішоходам. Тому даний тип пристрою вимагає високого рівня відповідальності від людини яка його використовує.

Гіроскутери порівняно з сігвеями мають деяку кількість переваг, поперше гіроскутери через наявність двох колісної системи потребують меншої кількості навичок в процесі керування оскільки від користувача вимагається тримати рівновагу лише в одній площині. Як і сігвеї гіроскутери мають досить невеликий ресурс використання, але за рахунок наявності двох мотор колеса з досить великим КПД модель потребує акумулятора меншої ємності що в свою чергу робить конструкцію легше і відповідно більш зручною в використанні та зберіганні. Так як і сігвеї гіроскутери представляють потенційну небезпеку у разі використання некваліфікованими водіями. Велика кількість даних транспортних засобів мають здатність до

Апаратне забезпечення та система управління для триколісного транспортного засобу максимальної швидкості в 20 км/год але трапляються варіанти коли данні моделі можуть здатність пересуватися з швидкістю більшою за 50 км/год, що в випадку аварії або іншої аварійно небезпечної ситуації представляє потенційну смертельну небезпеку для водія, інших учасників дорожнього руху або пішоходів.

Електросамокати порівняно з іншими транспортними засобами данного типу, мають ще більшу перевагу в використанні. Даний тип транспорту вимагає від користувача тримати рівновагу лише в одній площині, а наявність повноцінного руля робить керування ще більш зручним. Високоєфективні мотор-колеса та велика кількість вільного місця для встановлення акумулятора дають змогу даному виду транспорту працювати досить тривалий час. Через наявність великого запасу потужності та зручне керування, даний тип транспорту не рятує вбудоване обмеження швидкості, досить часто трапляються випадки коли користувачі вимикають обмежувачу і катаються на даному транспортному засобі зі швидкістю перевищуючу 70 км/год, що досить часто призводить до смертельних аварійних випадків. Оскільки в правилах дорожнього руху непередбачено обставини того що транспортний засіб який не підлягає обов'язковій реєстрації, та який не має двигуна внутрішнього згорання, може пересуватися з аварійно небезпечною швидкістю по тротуарам та дорогам загального вжитку, то водії даного транспортного засобу використовують його в небезпечних умовах як для себе так і для оточуючих, що є недопустимо.

В якості альтернативи розглядається електромобіль, на відміну від електросамокату електромобіль має кузов, систему пасивної безпеки можливість перевезення як людей так і вантажу до 1000 кг наявність 4 точок опору вимагає від водія лише необхідних навичок керування, необхідність тримати рівновагу взагалі не має значення. Електромобіль має потужний акумулятор з великою ємністю батареї яка дає змогу даному транспорту пересуватися на відстані від 200 до 500 км, деякі моделі електрокарів як на

Апаратне забезпечення та система управління для триколісного транспортного засобу приклад моделі які випускає такий виробник як Tesla Inc. Мають вбудовану систему автопілоту та дистанційне керування, що дозволяє використовувати даний тип транспорту з максимальною безпечністю для водія пасажира пішохода та інших учасників дорожнього руху. Але даний вид транспорту має і високу кількість недоліків. Через наявність акумулятора високої ємності його зарядка в домашніх умовах займає дуже велику кількість часу, зарядка яка надає необхідну потужність для швидкого завершення цього процесу знаходиться не в прямій доступності і має високу собівартість, використання електрокару передбачає що водій має відповідне водійське посвідчення що робить використання цього типу транспорту неповнолітнім особам просто неможливим. І найголовніше вартість, собівартість даного типу транспорту на відміну від інших наведених в сотні разів вища.

### **1.5.1 Компромісне рішення**

В якості альтернативного компромісного рішення необхідне створення моделі яка буде мати необхідну надійність конструкції, простоту в керуванні, конструкція повинна бути доступна для керування звичайному користувачу який зможе з відповідальністю підійти до своєї задачі, конструкція повинна мати системи безпеки базового рівня такі як пасок безпеки, мати акумулятор з відповідною ємністю яка дасть змогу керувати моделю в продовж певного часу. Модель повинна бути компактною, або мати змогу без складностей розбиратися на окремі складові для транспортування та подальшого зберігання в той час коли конструкція не використовується. Модель повинна мати систему авто-пілота, або систему яка дасть змогу керувати моделю дистанційно, та передавати зображення на персональний інструмент зв'язку та комунікації такий як смартфон. І головне, конструкція апаратного комплексу повинна мати не високу собівартість.

**Висновки до розділу 1**

В даному розділі було розглянуто принципи та основи вже створеного апаратного забезпечення.

На основі описаної інформації можна виявити найбільш необхідні якості які будуть притаманні створеному апаратному комплексу системи управління триколісного засобу. Найбільш високу увагу впродовж створення апаратного комплексу необхідно приділяти безпеці як водія так і пішохода.

Для того щоб комплекс мав певний ступінь конкурентоспроможності необхідно:

- Вирішити проблему довго тривалості роботи шляхом встановлення акумулятора з більшою ємністю ;
- Встановлення більш потужного акумулятора збільшить вагу моделі, щоб компенсувати це необхідно зробити відповідну для ваги комплексу систему амортизації;
- Необхідно вирішити проблему дистанційного керування та системи відеомоніторингу;
- Щоб вирішити проблему дистанції керування можна використовувати всесвітню мережу інтернет.



**РОЗДІЛ 2****РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ ПРОГРАМНО-АПАРАТНОГО  
КОМПЛЕКСУ****2.1 Огляд обладнання необхідного для створення апаратного комплексу**

Для створення апаратного комплексу необхідно забезпечити комплекс електродвигуном який дасть змогу комплексу пересуватися, необхідне джерело живлення для комплексу, необхідне апаратне забезпечення для керування електродвигуном комплексу та реалізації керування на дистанції та передачі відео зображення з моделі на пристрій яким буде здійснюватися керування, необхідно створення гальмівної системи комплексу, необхідне створення системи керування швидкості, системи передачі відеоряду за допомогою камер. Розглянемо детально характеристики основних компонентів які можна використовувати при реалізації схеми пристрою.

**2.1.1 Мікрокомп'ютер Raspberry Pi 4**

Raspberry Pi 4 – міні-комп'ютер з USB 3.0, Gigabit Ethernet і до 4 ГБ DDR4. Плата побудована на базі SoC Broadcom BCM2711, яка містить чотири 64-розрядних ядра ARM Cortex-A72 з тактовою частотою 1.5 ГГц. На відміну від будь-якої попередньої плати, Raspberry Pi 4 Model B доступна в трьох різних моделях, кожна з яких пропонує різні варіанти пам'яті з 1 ГБ, 2 ГБ, 4 ГБ або 8 ГБ LPDDR4 SDRAM оперативної пам'яті.

Найбільш помітне відміну від попередніх моделей полягає в тому, що на новій платі відсутня мікросхема LAN7515, яка виконувала функції USB і Ethernet. На його місці VLI VL805, яка забезпечує USB 3.0 Hub через шину PCI Express. Використання шини PCI Express, що надається новим BCM2711, означає, що тепер є не тільки можливість USB 3.0, але і Gigabit Ethernet, який тепер надається з використанням Broadcom BCM54213PE окремо від USB. Бездротовий зв'язок забезпечується тим же чіпом Cypress CYW43455, який ми бачили на Raspberry Pi 3 модель B +. Пропонується двухдиапазонная

Апаратне забезпечення та система управління для триколісного транспортного засобу бездротова мережа IEEE 802.11.b / g / n / ac 2,4 ГГц і 5 ГГц, а також Bluetooth 5.0 і Bluetooth LE.

Ще одним великим відмінністю є роз'єм живлення, відсутній роз'єм micro-USB від попередніх моделей, а на його місці роз'єм USB-C. Raspberry Pi 4 Model B може знадобитися до 3 А, що не могло забезпечити попередній джерело живлення через micro-USB. 4-я модель може харчуватися і від джерела постійного струму 5 В через роз'єм GPIO і, як і Raspberry Pi 3 Model B +, від Power over Ethernet (PoE) з використанням офіційної PoE HAT, випущеної раніше поряд з попередньою моделлю.

Мікрокомп'ютер Raspberry Pi 4 приведено на рис. 2.1, а основні характеристики – в табл. 2.1.



Рисунок 2.1 – Розташування виводів мікроконтролера ATmega8A

Таблиця 2.1 – Основні характеристики мікроконтролера ATmega8A

| Параметри                           | Значення  |
|-------------------------------------|---|
| Напруга живлення<br>(номінальна), В | 4,5 – 5,5   |
| Процесор                            | 64-бітний чотирьохядерний ARMv8 Cortex-A72 процесор з тактовою частотою 1.5 ГГц |
| Графічний процесор                  | VideoCore VI GPU підтримує  |

|                 |  |
|-----------------|--|
|                 | стандарти OpenGL ES 3.0; кодирование H.264 до 1080p30, декодирование 4Kp60 видео (H.265 до 4Kp60, H.264 до 1080p60)  |
| ОЗУ             | 1/2/4/8 ГБ SDRAM LPDDR4  |
| Сховище         | слот для карти пам'яті MicroSD   |
| Ethernet        | 10/100/1000 Мбит Gigabit Ethernet (контроллер Broadcom BCM54213PE)   |
| ОС              | Ubuntu, Debian, Fedora, Arch Linux, Gentoo, RISC OS, Android, Firefox OS, NetBSD, FreeBSD, Slackware, Tiny Core Linux, Windows 10 IOT  |
| Wi-Fi/Bluetooth | 2.4 ГГц і 5 ГГц IEEE 802.11.b / g / n / ac WI-FI і Bluetooth 5.0 Low Energy (BLE), що забезпечуються мікросхемою Cypress CYW43455  |
| Розміри         | 85.6 мм x 56.5 мм x 17 мм  |
| Вага            | 45 грам  |
| Периферія       | 40 портів введення-виведення загального призначення (GPIO), UART (Serial), I <sup>2</sup> C / TWI, SPI з селектором між двома пристроями; Піни харчування: 3,3 В, 5 В і земля. |

Також змінено відео порти. Відсутня повнорозмірне гніздо HDMI, на його місці два порти мікро-HDMI, це означає, що новий Raspberry Pi 4 Model B має підтримку двох моніторів - одного екрану в 4K зі швидкістю 60 кадрів в секунду або двох екранів 4K зі швидкістю 30 кадрів в секунду. Поруч з двома портами мікро-HDMI є 4-полюсний стереовиход і композитний відеороз'єм.

Поряд з великими змінами деякі речі не змінилися. Зберігання і раніше забезпечується картою Micro-SD з роз'ємом в звичному місці на нижньому боці плати. Як і раніше підтримує роз'єм порту дисплея MIPI DSI і роз'єм камери MIPI CSI, а також стандартний 40-контактний роз'єм GPIO. Однак є деякі зміни для 40-контактного роз'єму, хоча він все ще сумісний. Роз'єм нового Raspberry Pi має підтримку додаткових роз'ємів  $4 \times \text{UART}$ ,  $4 \times \text{SPI}$  і  $4 \times \text{I2C}$ .

Даний мікрокомп'ютер зможе реалізовувати передачу відеоряду на пристрій який в свою чергу зможе передавати сигнали керування на мікрокомп'ютер, який в свою чергу зможе реалізовувати керування моделлю.

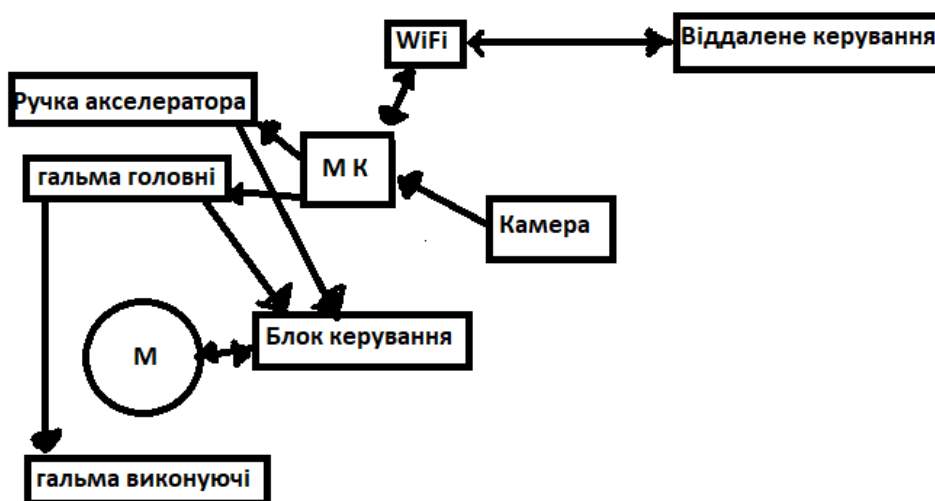


Рисунок 2.2 – Структурна схема роботи комплексу

Як можна бачити з схеми (Рис.2.2) Мікрокомп'ютер (МК) отримує дані з камери які в свою чергі відсилає на модуль мережевого зв'язку WiFi який в свою чергу відсилає сигнал на пристрій відтворення відео зображення, подальше керування знову відсилається на мікрокомп'ютер який оброблює сигнал і відсилає відповідні команди на органи управління такі як ручка акселератора та ручку головних гальм, з ручок керування сигнал відсилається на блок керування у випадку гальм механічно відсилається сигнал керування гальмами в той час як блок керування оброблюючи сигнали гальмівної ручки та ручки акселератора приводить двигун в дію або навпаки вводить двигун в режим роботи генератора (рекуперація).

### 2.1.2 Електро двигун

Безколекторний або як його ще називають Бесщеточний мотор серії VEOLA XF40 Даний вид моторів застосовується на електровелосипедах.



Рисунок 2.3 – Мотор VEOLA XF40

На Рис. 2.3 зображено безколекторний двигун VEOLA XF40. Мотор-колесо є агрегат, що об'єднує колесо і вбудовані в нього тяговий електродвигун, силову передачу і, в деяких випадках, гальмівну систему (отже, кожне мотор-колесо має індивідуальний привід). Встановлюється, як правило, у підвішеному до рами кронштейні (у разі коли колесо не є керованим) або у встановленому в поворотній цапфі підшипнику (у випадку коли колесо є одночасно провідним і керованим). Живиться енергією від двигуна внутрішнього згоряння через електромеханічну трансмісію (переважно на автомобільній техніці, головним чином важкої), від контактної мережі (на тролейбусах та тролейвозах) або від акумулятора (на електромобілях та електровелосипедах, або, як додаткове джерело енергії, на автомобільній двигуном внутрішнього згоряння, такий як гібридні автомобілі, або тролейбуси). Існує два режими роботи мотор-колеса - тяговий та генераторний. У тяговому режимі обертання передається з валу якоря електродвигуна, що працює в руховому режимі через редуктор до

Апаратне забезпечення та система управління для триколісного транспортного засобу внутрішнього зубчастого вінця ведучого колеса; у генераторному режимі, що використовується для електричного гальмування, електродвигун переходить у генераторний режим роботи, а електроенергія перетворюється на тепло на гальмівному реостаті (реостатне гальмування) або повертається в електричну мережу або застосовується для заряджання акумуляторів (рекуперативне гальмування)

Технічні характеристики мотор-колеса:

- Мотор: безредукторний, без щітковий
- Вольтаж: 48 Вольт,
- Потужність: 500 Ватт,
- Ефективність:  $\geq 80\%$
- Максимальна температура: 150 °C
- Гучність: <60 дБ,
- Захист від вологи: IP54,
- Кількість зірок: 1-6
- Гальмо: ободне/дискове,
- Колір чорний
- Виробник: MXUS
- Вага двигуна-колеса: 5,8 кг.

Посадкові розміри:

- Довжина та діаметр осі: 218 мм; M14
  - Довжина різьбової частини ліворуч: 32 мм
  - Довжина різьбової частини праворуч: 38 мм
  - Посадкова ширина передньої вилки: 136 мм
  - Довжина кабелю для підключення контролера: 0,8м
- Комплектація:  
див. фото.
- Розміри: 27,8 × 27мм.
  - Вага: 48 г..

### 2.1.3 Регулятор – Акумулятор

Регулятор-Акумулятор RE-48V 10,4Ah – це доступний контролер для безколекторних електромоторів з вбудованим акумулятором з пакету акумуляторів 18650, який часто встановлюється на електросамокатах, електровелосипедах та іншому.



Рисунок 2.4 – Регулятор безколекторний RE-48V 10,4Ah

На рис. 2.4 Зображено безколекторний регулятор RE-48V 10,4Ah.

Регулятор не має можливості зміни будь-яких параметрів настройки. Ні за допомогою радіопередавача (по піках), ні за допомогою програмної карти.

Завдяки високоточній кварцовим генератором, температура не впливає на робочий діапазон ШІМ. Регулятор швидкості 30 А володіє супервисокою частотою оновлення (490 Гц), без буферизації вхідного сигналу, в результаті чого має високу швидкість реакції.

Специфікація:

- Номінальна напруга: 48 В
- Номінальна ємність(25 С±0,5С): 10,4 Ач
- Вес (приблиз.): ≤4.5кг
- Розмір (Д\*Ш\*В): 361\*129\*85 (мм)

Характеристика розряду:

- Макс. струм розряду: 25 А
- Піковий струм розряду (500 ms): 80 А

Апаратне забезпечення та система управління для триколісного транспортного засобу

- Напруга відсічення: 39 В±1 В

Характеристика заряду:

- Напруга заряду: 54,6±0,2 В
- Рекомендований струм заряду: 2 А
- Година заряду: 4-6 годин

Опір

- Внутрішній опір: ≤120mΩ
- Саморозряд: ≤2% на міс.

Основні характеристики

- Елементи: 18650-3,6V 2,6 Ah
- BMS: 25A-80A
- Піковий струм: 40А (10сек)
- Живлення: 2-3S,
- Регулятор напруги вбудований ВЕС: 5V / 3A
- Колір червоний
- Розмір: 50 x 23 x 8mm / 2.0 x 0.9 x 0.3in
- Вага: 28г

#### **2.1.4 Серводвигун**

Серводвигун - це привід, вал якого може встати в задане положення або підтримувати задану швидкість обертання. Іншими словами, валом сервоприводу можна управляти, наприклад, задаючи йому положення в градусах або певну частоту обертання.





Рисунок 2.4 – Серводвигун Emax ES09MD

Керування серводвигуном відбувається за допомогою шим сигналу, за рахунок зміни частоти шим сигналу змінюється кут повороту серводвигуна.

Характеристики серводвигуна Emax ES09MD:

- Швидкість перекладки - 5 Вольт 0.10 сек;
- 6 Вольт 0.09 сек;
- Зусилля 5 Вольт 2.2 Кг / см;
- 6 Вольт 2.5 Кг / см.;
- Струм спокою 12мА;
- Струм перекладки із середньою швидкістю без навантаження близько 120мА;
- Струм споживання на упорі (2.2кг / см, 5 Вольт) 1А.

## 2.1.4 Камера

В якості Камери використовується веб камера ASUS Webcam C3



Рисунок 2.5 – Веб камера ASUS Webcam C3

Веб камера має високу чіткість (Full HD, 1920 x 1080) та плавність (30 кадрів/с) відеотрансляції. Ефективне фільтрування навколишніх шумів для чіткої комунікації. Фіксатор з можливістю регулювання кута нахилу до 90 ° для встановлення камери на дисплеї ноутбука або монітора, а також механізм обертання на 360 ° по горизонталі.

Широкоформатний об'єктив – ідеально підходить для конференцій, дистанційної роботи або чатів із двома або більше учасниками, що сидять поруч.

## 2.1.5 Гальмівна система

В якості Гальмівної системи використовується гальмівна система на базі дискового гальмівного принципу. З назви зрозуміло, що для гальмування використовується гальмівний диск, на велосипедах він називається ротором. Виготовляються вони з твердих сплавів сталі. Хоча є варіанти з нержавіючої сталі, наприклад shimano rt56 (на картинці найлівіший, rt-ротор), тому він один з найдешевших і підходить тільки для органічних (resin) колодок. Ротора можуть мати різний дизайн, але, виходячи з практики на гальмування, це не впливає. Також існують ротори на алюмінієвому павуку (2 та 3 катинки). Павук дозволяє ротору швидше охолоджуватися і захищає його від

Апаратне забезпечення та система управління для триколісного транспортного засобу деформацій через перегрівання. Форми павуків та їх колір значення не мають. На правій картинці "плаваючі" ротора від відомого виробника hope, вони досить дорогі і красиві, але у них буває один недолік - вони можуть розбвтатися і почати люфтити.



Рисунок 2.6 – Різновиди гальмівних дисків.

Існує 2 стандарти кріплення роторів до втулки: 6-bolt (Рис. 2.6 ліворуч) та center-lock (CL) (Рис. 2.6 праворуч). Не можна сказати, який з них краще. Центрлок був придуманий компанією шимано скоріше з комерційною метою (кіпі втулку і ротор обов'язково у нас). У CL іноді виникає люфт і його не можна зняти без спеціального колюча (як для касети), проте ротора міняти доводиться рідко. На картинці ротора та втулки під обидва стандарти. Там де CL на роторі одягнена кришечка, під нею зубці, як на втулці.

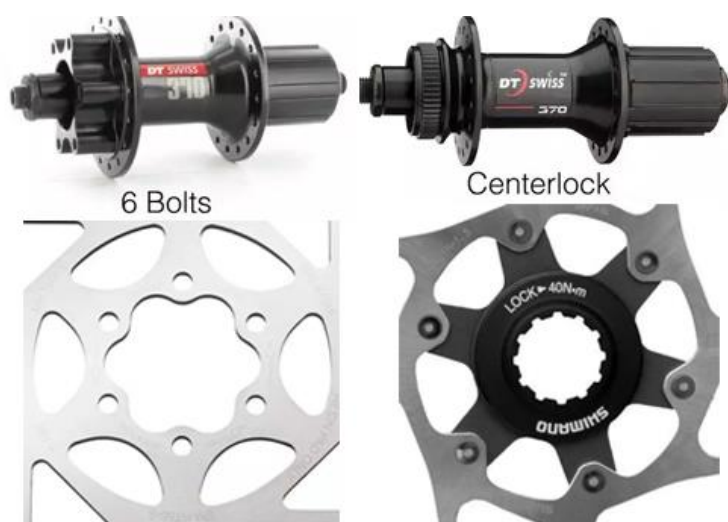


Рисунок 2.7 – Різновиди роторів гальмівних дисків

Різниця між роторами різних виробників(рис. 2.7) практично немає.

Різниця в гальмуванні є тільки з роторами з нержавіючої сталі, інші

Апаратне забезпечення та система управління для триколісного транспортного засобу приблизно однакові. Існує кілька стандартних розмірів роторів(в мм): 140, 160, 180, 200, 203. Найпоширеніший - 160. Великі ротори використовуються в агресивних дисциплінах, оскільки мають більший важіль і краще охолоджуються. Не на будь-який велосипед можна поставити великі ротори, бо вони можуть вбити велосипед. Наприклад якщо на вилку ставити ротор більше заявленого максимуму, коли вона швидко розхитається і зламається, ззаду можуть тріснути пір'я. У міському застосуванні, прогулянковому та кросскантрі різниці між роторами на павуці і без ні, так само як і різниці у розмірах. Старші ротора шимано RT-8x і RT-9x зроблені за технологією Ice-Tech, суть у тому, що поверхня, що гальмує, зроблена як бутерброд, в середині алюміній, а по краях сталь (поверхня тертя з колодками), а RT-99 має ще й радіатор, але у місті все це не потрібно, тільки гроші віддасте. Будьте обережні з китайськими роторами та роторами з усім відомого сайту. Так, вони красиві, але бувають проблемні, наприклад, можуть розірватися або може відвалитися павук.

### 2.1.6 Рама конструкції

Рама конструкції виготовлена з профільної труби товщина стінки якої 1.5мм, труба має квадратний формат з параметрами 40×40 мм., та труба з параметрами 20×40 мм. На рис.2.8 приведено модель триколісної системи.



Рисунок 2.8 – Триколісна система пересування

## 2.2 Опис функціонування схеми

Після огляду характеристик компонентів, що можуть бути використаними у складі пристрою, було спроектовано принципову схему. Схема розроблена з урахуванням специфічних вимог, що визначаються режимами керування пристрою.

Структурну схему програмно-апаратного комплексу приведено на рис. 2.9, а повну принципову схему – в додатку А.

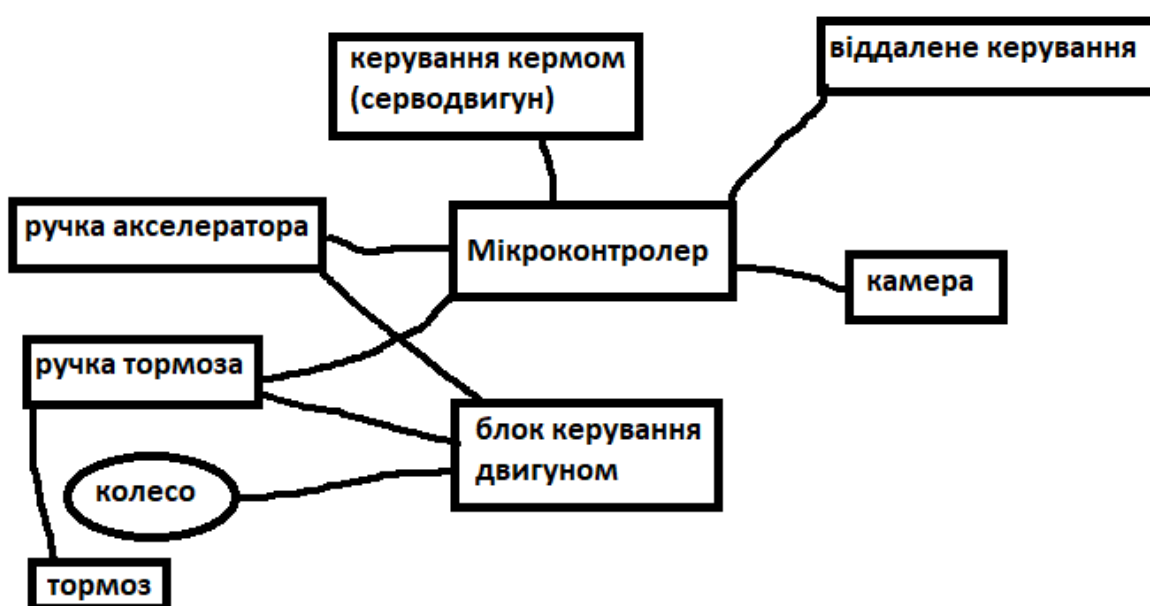


Рисунок 2.9 – Структурна схема програмно-апаратного комплексу

Схема складається з основного мікроконтролера до якого підключені периферійні пристрої для зчитування та перетворення інформації в дію.

До кожного периферійного пристрою використовується певні інтерфейси підключення та певні схеми керування. Серводвигуни та мотори керуються за допомогою ШИМ-сигналу, для керування серводвигунів та моторів використовуються різні ШИМ-сигнали але за своєю основою вони досить подібні

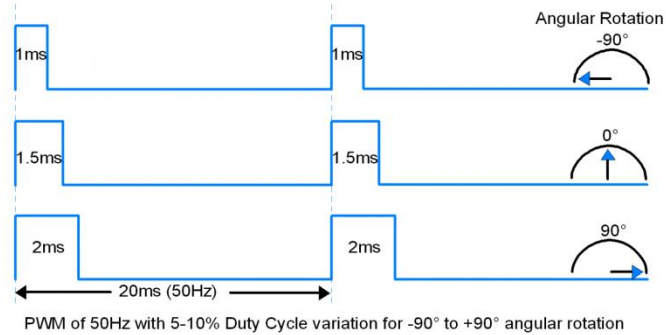


Рисунок 2.10 – ШИМ-сигнали керування

На Рис. 2.10 зображено відношення шим сигналу до положення сервопривода, для керування безколекторного мотору використовується шим-сигнал в 1мс. для положення нерухомості двигуна та 2мс. для його максимальних оборотів. Для інших модулів таких як веб камера використовується інтерфейс usb.

### 2.3 Опис функціонування окремих частин схеми.

Схема керування апаратного комплексу має складну систему підтримки роботи комплексу. Весь комплекс може керуватися в 2 режимах ручному та віддаленому.

#### 2.3.1 Ручний режим керування

Під час ручного режиму керування система дистанційного керування може знаходитися в неактивному або псевдо активному положенні. Керування відбувається за рахунок взаємодії з важелем акселератора. Акселератор являє собою резистор який змінює свій опір в залежності від натискання на важіль, опір зчитується блоком керування двигуна який відсилає потужний сигнал керування на головний двигун, у випадку коли важіль гальм знаходиться в активному положенні блок керування двигуном переводить двигун в режим рекуперації, тобто обертальний момент перетворюється на електрорушійну силу (режим рекуперацію) і тим самим під заряджає акумулятор.



### 2.3.2 Режим віддаленого використання

Для переходу в режим віддаленого використання в мобільному додатку треба відсилати відповідні сигнали на апаратний комплекс, веб інтерфейс має зрозумілий інтуїтивно стан, тож керування не буде заздавати незручностей. Веб інтерфейс дає змогу отримувати зображення з камери встановленій на триколісному комплексу.

В той час на апаратному комплексі гальма реалізуються за рахунок серводвигуна який приводить в дію механізм важеля за допомогою системи редукторів по принципу роботи подібні до електронної системи стоянкового гальма автівки(рис. 2.11).



Рисунок 2.11 – серводвигун стоянкового гальма

Імітація роботи важеля акселератора реалізується за допомогою зміни опіру в мікроконтролері який підключен до блоку керування двигуном. Найбільш складним є елемент керування рулем (рис. 2.12) оскільки для керування рулем необхідно докладати певних зусиль прямою передачею обертального моменту цього не досягти, тому для того щоб збільшити силу з якою механізм рушиться використовується потужний серводвигун з редуктором.

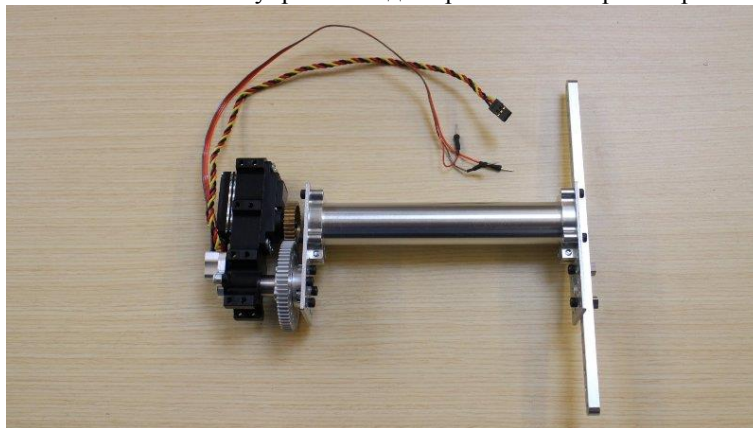


Рисунок 2.11 – серводвигун руля

## Висновки до розділу 2

В даному розділі було розглянуто апаратну реалізацію пристрою для Керування та стабілізації. Приведено опис основних використаних компонентів, їх характеристики, та аргументовано їх вибір.

Після ознайомлення з характеристиками компонентів було спроектовано принципову схему програмно-апаратного комплексу, використовуючи стандартні широко доступні компоненти, приведено опис принципу її функціонування та призначення окремих її частин.

Розробка схеми проводилась з врахуванням можливості подальшого розширення функціоналу програмно-апаратного комплексу без суттєвих змін в розробленій схемі. Передбачена можливість майбутньої опції автокерування.



## РОЗДІЛ 3

### РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ ПРИСТРОЮ

#### 3.1 Опис середовища розробки Arduino IDE

При розробці програмної частини програмно-апаратного комплексу було вирішено використати мову програмування C, яка буде імпортована на мову програмування Python альфа версія для керування та стабілізації була розроблена для платформи Arduino в середовищі розробки (IDE) Arduino, скріншот стартової сторінки якого показано на рис. 3.1.

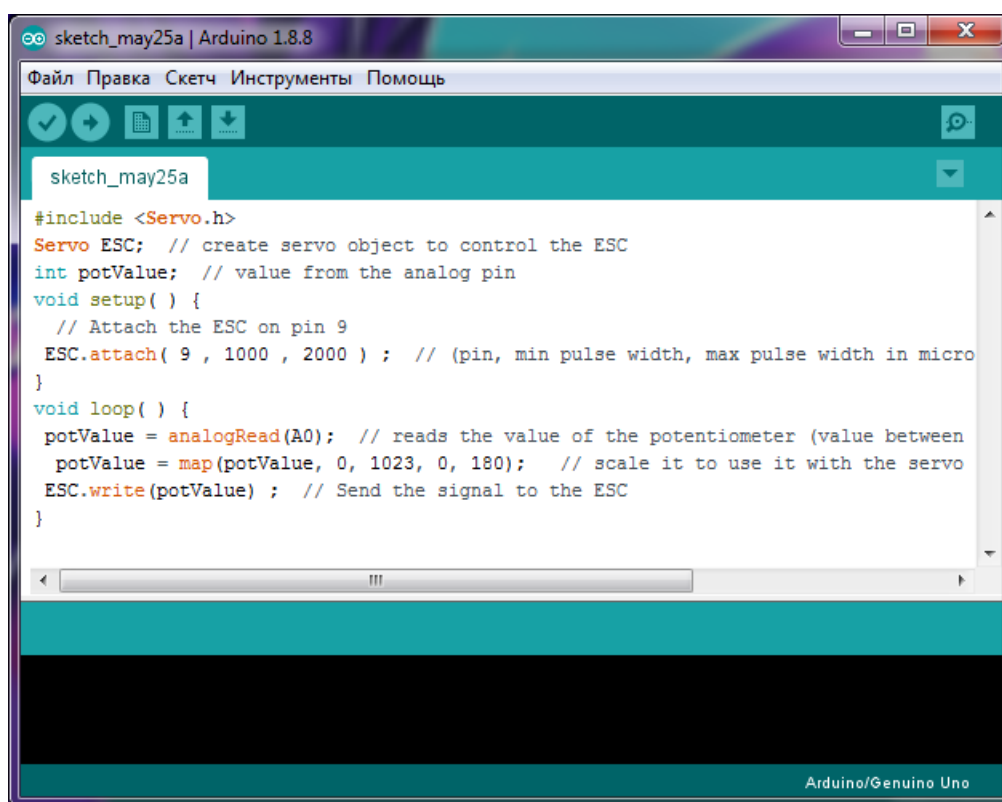


Рисунок 3.1 – Стартова сторінка IDE Atmel Studio

Arduino IDE – це IDE, розроблена фірмою Arduino Software для підтримки їх продукції – мікроконтролерів сімейств IA-32, x86-64, ARM. Дана IDE підтримує мови програмування C, C++ та мову асемблера вказаних типів мікроконтролерів. Код для срібних IDE випущений під загальнодоступною ліцензією GNU версії 2. Arduino IDE надає бібліотеку програмного забезпечення з проектом проводки, яка пропонує багато загальних процедур введення та використання. Для написаного користувачем

Апаратне забезпечення та система управління для триколісного транспортного засобу коду потрібні лише дві базові функції для використання ескізу та основного циклу програм, які складаються із загальнодоступними програмами `main ()` у виконавчій програмі GNU, також включені в дистрибутив IDE.

Arduino IDE використовує програму `avrdude` для перетворення виконавчого коду в текстовому файлі в шістнадцятеричній кодировці, який завантажується в плату Arduino програмою-завантаженням у вбудованому програмному забезпеченні плати. За замовчуванням, у своїх інструментах для завантаження кода користувача на офіційних платах Arduino використовує `avrdude`. З різними популярностями Arduino інші поставники в якості програмної платформи почали вводити користувальницькі компілятори та інструменти з відкритим вихідним кодом (ядра), які створюють і завантажують ескізи в інших мікроконтролерах, не підтримуючи офіційної лінійної мікроконтролерів Arduino.

### 3.2 Налаштування Raspberry Pi

Оскільки в якості апаратної платформи встановлено Raspberry Pi 4 в якості операційної систему було вибрано Armbian Linux.

Armbian - це дистрибутивний Linux, доступний для самих різних пристроїв ARM: Orange Pi, Banana Pi, Odroid, Olimex, Cubieboard, Roseapple Pi, Pine64, NanoPi. Він заснований на Ubuntu та / або Debian. Дана операційна система доступна як безплатна операційна система з вільним доступом. Для встановлення цієї операційної системи необхідно встановити операційну систему на комп'ютер для від лагодження та встановити її на `micro sd card`, потім `micro sd card` необхідно встановити в Orange Pi. При першому включенні системи необхідно під'єднати мікрокомп'ютер до роутера та ввімкнути систему, в налаштуванні роутера подивитися який статичний IP отримав мікрокомп'ютер. Для подальшої роботи нам необхідно програмне забезпечення PuTTY.

PuTTY - це клієнт SSH і telnet, який спочатку розробляє для платформ Windows. Безкоштовно розповсюджується клієнт для різних протоколів удаленого доступу, включаючи SSH, Telnet, rlogin

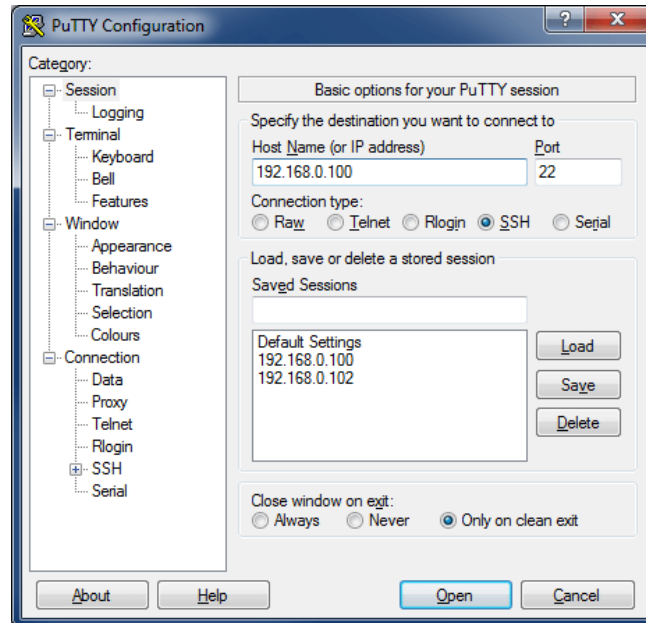


Рисунок 3.2 – Вікно середовища PuTTY

На рис. 3.2 зображено середовище PuTTY в якому необхідно налаштувати SSH підключення через порт 22 до мікрокомп'ютера Raspberry Pi при першому підключенні мікрокомп'ютер потребує зміни пароля при підключенні.

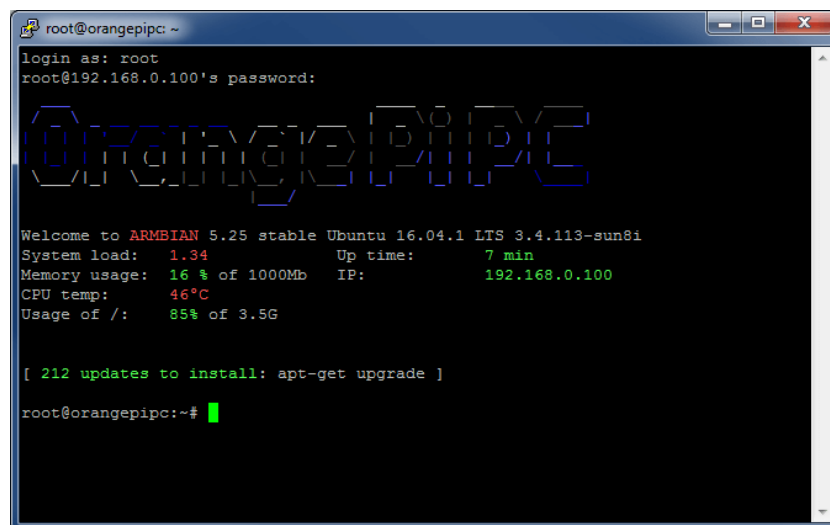


Рисунок 3.3 – Налаштування першого підключення

На рис. 3.3 зображено налаштування системи Raspberry Pi при першому підключенні мікрокомп'ютера, для подальшого більш комфортного використання мікрокомп'ютера необхідно прописати статичний IP для підключення до локальної мережі за допомогою інтерфейсу WI-Fi.

Подальша робота з прототипом реалізується за допомогою онлайн компілятора Python – repl.it. Програмний код який був вдало скомпільований імпортується через SSH на мікрокомп'ютер.

### 3.2 Налаштування підключення через WiFi

Для підключення інтернет сполучення необхідно використовувати або локальні мережі WiFi які працюють на території де буде працювати комплекс триколісної моделі, або підключитися до мобільного телефону водія який роздає мережу WiFi.

За допомогою такого функціоналу можна реалізувати передачу та отримання сигналів керування та відео потоку через Internet.

Отже ми маємо вихід до мережі але нам необхідно налаштувати веб сервер для обробки запитів та ретрансляції відео потоку. Є декілька варіантів вирішення цієї проблеми, купити VPS / VDS сервер. VPS / VDS - віртуальний сервер, який створюється засобами віртуалізації на фізичному сервері, який в свою чергу знаходиться в дата-центрі провайдера.

Кожен віртуальний розділ - це ізольована середовище та робота з нею аналогічна роботі з віддаленим фізичним сервером. Підключення йде по протоколу rdp (Windows Server OS) або ssh (сімейство ОС Linux).

На VPS даються повні адміністративні (root) права, можна встановлювати будь-які програми, бази даних. Число сайтів на 1 віртуальному сервері необмежена, і залежить тільки від величини, що зберігається. Конфігурацію віртуальних серверів зазвичай можна збільшити або зменшити залежно від провайдера, або прямо в особистому кабінеті хостера, або при зверненні в тих підтримку. Інший варіант створення сервера

Апаратне забезпечення та система управління для триколісного транспортного засобу на виділеному комп'ютері. Для ретрансляції відео потоку зручним рішенням є використання WebRTC (англ. Real-time communications - комунікації в реальному часі) - проект з відкритим вихідним кодом, призначений для організації передачі потокових даних між браузерами або іншими підтримують його додатками за технологією точка-точка. В якості веб сервера було вирішено використовувати веб сервер Apache HTTP-сервер який встановлен на Orange Pi, тобто одноплатний комп'ютер має повноцінне використання як мультимедійний пристрій.

Apache HTTP-сервер – Відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та других операційних систем.

Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation. Користувачі вводять однаковий покажчик ресурсів (URL) для визначення інтернет-сервера за його повністю кваліфікованому доменному імені (FQDN) та шляхи до необхідного ресурсу. Наприклад, щоб побачити домашню станицю інтернет-сайту Ubuntu, користувач повинен ввести тільки FQDN: [www.ubuntu.com](http://www.ubuntu.com)

Для перегляду сторінки про співтоваристві користувач повинен ввести наступний шлях: [www.ubuntu.com/community](http://www.ubuntu.com/community)

Найбільш поширений протокол, який використовується для передачі інтернет-сторінок, - це гіпертекстовий протокол передачі (HTTP). Такі протоколи, як HTTP поверх SSL (HTTPS) і протокол передачі файлів (FTP) - протокол для завантаження і отримання файлів, також використовуються.

### **3.3 Програмна частина керування**

Отже для керування апаратною платформою нам необхідна платформа для керування яка є додатком для смартфона. Для створення додатку використовується програмне середовище Android Studio. Android Studio - інтегроване середовище розробки (IDE) для роботи з платформою Android, анонсована 16 травня 2013 року на конференції Google I / O.

Апаратне забезпечення та система управління для триколісного транспортного засобу

Дана IDE перебувала у вільному доступі починаючи з версії 0.1, опублікованій в травні 2013, а потім перейшла в стадію бета-тестування, починаючи з версії 0.8, яка була випущена в червні 2014 року. Перша стабільна версія 1.0 була випущена в грудні 2014 року, тоді ж припинилася підтримка плагіна Android Development Tools (ADT) для Eclipse.

Android Studio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, - офіційне засіб розробки Android додатків [2]. Дане середовище розробки доступна для Windows, macOS і Linux [3]. 17 травня 2017, на щорічній конференції Google I / O, Google анонсував підтримку мови Kotlin, використовуваного в Android Studio, як офіційної мови програмування для платформи Android на додаток до Java і C ++.

### 3.4 Опис користувацького інтерфейсу та тестування пристрою

Користувацький інтерфейс (рис 3.4) має в наявності декілька кнопок та повзунків за допомогою яких реалізується керування



Рисунок 3.4 – Користуватський інтерфейс

На головному екрані є декілька кнопок та повзунків, за допомогою кнопок з стрілками наліво та вправо відповідно реалізується керування кермом в відповідних сторонах, серед кнопок є кнопка головного меню в меню є опція зміни яскравості та контрасту зображення серед функціоналу головного екрану є 2 повзунки, перший відповідає за швидкість пересування моделі, а другий відповідає за тормозну силу, при простому натиску на

Апаратне забезпечення та система управління для триколісного транспортного засобу повзунк на апаратному комплексі реалізується режим рекуперації, при взаємодії з повзунком сила тормозної дії збільшується.

### **Висновки до розділу 3**

В даному розділі розглянуто деталі реалізації програмної частини програмно-апаратного комплексу. Розробка програмної частини проводилась в середовищі розробки Arduino IDE на мові програмування C в подальшому імпортованої в Ruytron. Вибір даної мови програмування пов'язаний з тим, що її використання дозволяє максимально повно використати можливості периферійних пристроїв мікроконтролера та швидкодію процесора.

При проектуванні програмної частини найбільша увага зверталась на реалізацію модульної структури програмного коду та можливість розширення функцій та можливостей пристрою.

## РОЗДІЛ 4

### ТЕСТУВАННЯ ПРИСТРОЮ, ТА ІНСТРУКЦІЯ З ВИКОРИСТАННЯ

#### 4.1 Тестування пристрою

Тестування пристрою проходить в декілька етапів які він повинен пройти без виявлених несправностей як в механічній так і в програмних частинах, тестування буде проводитись як з механічним агрегатом так і з його апаратним забезпеченням.

Перед початком тестування механічної частини комплексу, необхідно впевнитись в повній придатності апаратного забезпечення комплексу, коректної роботи серверу для віддаленого керування та працездатності апаратної частини забезпечення руху та гальмування.

##### 4.1.1 Тестування двигуна та інвертора

Для тестування та діагностики електродвигуна був взят принцип перевірки асинхронних двигунів промислового вжитку, за внутрішньою будовою не відрізняються від того що використовується під час побудови комплексу.

##### 1. Вхід живлення

Аналіз електроживлення, що надходить на електропривод, є ефективною першою дією для визначення наявності в ланцюгу живлення спотворень, перешкод або шумів, які можуть вплинути на заземлення. Перевірки Порівняйте номінальну напругу приводу з фактичною напругою, що подається, щоб швидко визначити, чи знаходяться значення в допустимих межах. Якщо вихід за межі діапазону становить більше 10%, це може свідчити про проблеми з напругою живлення. Переконайтеся, що вхідний струм знаходиться в межах максимально допустимого номіналу, а провідники мають відповідний розмір.

Порівняйте виміряне значення частоти із заданим значенням. Різниця, що становить понад 0,5 Гц, може призвести до проблем.



Переконайтеся, що гармонійне викривлення знаходиться в межах допустимого рівня. Візуально перевірте форму сигналу або перегляньте екран гармонічного спектру, на якому показано як загальне гармонічне спотворення, так і окремі гармоніки. Наприклад, форми сигналу з плоскою вершиною можуть свідчити про нелінійне навантаження, підключеного до того ж ланцюга живлення. Якщо загальне гармонічне спотворення (THD) перевищує 6%, це свідчить про наявність потенційної проблеми.

Перевірте асиметрію напруги на вхідних клеммах, щоб переконатися, що асиметрія фаз не надто висока (менше 6–8 %), і що чергування фаз є правильним. Високе значення асиметрії напруги може вказувати на урвище фази. Показання, що перевищує 2%, може призвести до переривання напруги та спрацьовування системи захисту приводу від навантаження або порушити роботу іншого обладнання.

Перевірка асиметрії струму. Надмірна асиметрія може вказувати на несправність випрямляча приводу. Асиметрія струму більше 6% може вказувати на несправність перетворювача електроприводу та призвести до потенційних проблем.

## 2. Шина постійного струму

Перетворення змінного струму на постійний у приводі має велике значення. Правильну напругу та відповідне згладжування з низьким рівнем пульсацій необхідно для забезпечення максимальної продуктивності приводу. Високий рівень пульсації напруги може бути ознакою несправності конденсаторів або некоректного визначення розмірів підключеного електродвигуна. Функцію запису аналізатора електроприводів Fluke серії MDA-500 можна використовувати для динамічної перевірки продуктивності постійного струму шини в робочому режимі з навантаженням. Як альтернативу для виконання цієї перевірки можна використовувати вимірювальний прилад Fluke ScopeMeter® або вдосконалений мультиметр.

### Перевірки

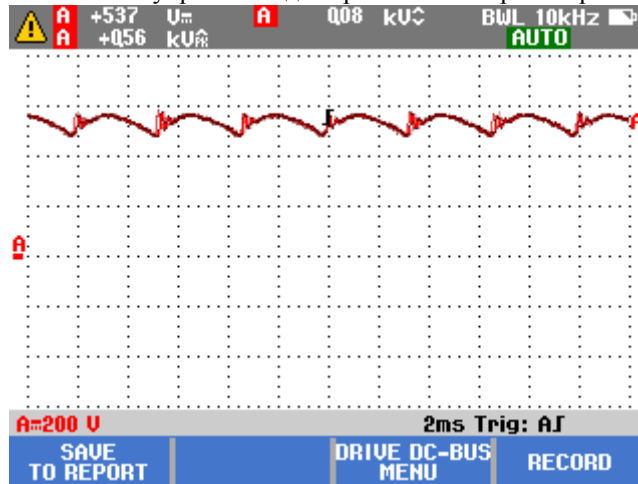


Рисунок 4.1 – Пульсація на шині постійного струму

Визначте, чи напруга шини постійного струму є пропорційним піковому значенню вхідної напруги лінії. За винятком керованих випрямлячів, напруга повинна бути кратна 1,31-1,41 середньоквадратичного значення напруги лінії. Низькі показники напруги постійного струму можуть призвести до спрацювання приводу, що може бути викликано низькою вхідною напругою мережі або будь-яким перекручуванням вхідної напруги, наприклад, спотворенням плоскою вершиною.

Перевірте наявність будь-яких спотворень чи помилок у піковій амплітуді напруги лінії. Це може призвести до помилки, пов'язаної з підвищеною або зниженою напругою. Показ напруги постійного струму  $\pm 10\%$  від номінальної напруги може свідчити про наявність несправності.

Визначте, чи піки пульсації змінного струму мають різний рівень повторень. Після перетворення змінного струму на постійний на шині постійного струму залишатиметься невелика складова пульсації змінного струму. Напруги пульсації вище 40 можуть бути викликані несправністю конденсаторів або недостатнім номіналом приводу для підключеного електродвигуна або навантаження.

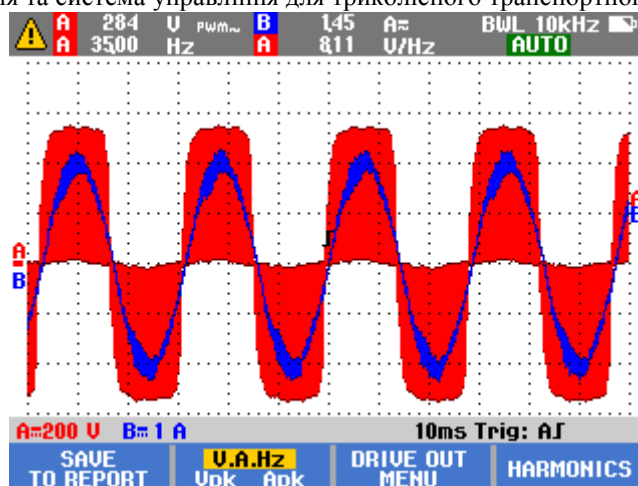


Рисунок 4.2 – Напруга та струм на виході привода

### 3. Вихід приводу

Перевірка на виході приводу має велике значення для забезпечення правильної роботи електродвигуна і допоможе у вирішенні проблем, що виникають у ланцюгах приводу. Перевірки

Переконайтеся, що напруга та струм знаходяться у відповідних межах. Через високий вихідний струм електродвигун може перегріватися, що скорочує термін служби ізоляції статора.

Переконайтеся, що відношення напруги до частоти (В/Гц) знаходиться в межах встановленого діапазону електродвигуна. При високому відношенні електродвигун може перегрітися, при низькому відношенні відбудеться зниження моменту електродвигуна, що крутить. Стабільне значення частоти та нестабільне значення напруги можуть свідчити про несправність шини постійного струму; нестабільне значення частоти та стабільне значення напруги можуть свідчити про проблеми перемикачів (БТІЗ). Нестабільні значення частоти та напруги свідчать про потенційні проблеми з ланцюгами регулювання швидкості.

Перевірте вихідну потужність приводу, звертаючи увагу на відношення напруги до частоти (Н/Ч) та на модуляцію напруги. При високому співвідношенні напруги/частоти електродвигун може перегрітись. При низькому відношенні Н/Ч підключений електродвигун може не

Апаратне забезпечення та система управління для триколісного транспортного засобу  
забезпечувати момент, що крутить, під навантаженням, необхідний для ефективного виконання заданого процесу.

Перевірте модуляцію напруги, використовуючи вимірювання між фазами. Високі піки напруги можуть пошкодити ізоляцію обмотки електродвигуна та призвести до спрацьовування приводу. Піки напруги вище 50% від номінальної напруги свідчать про наявність несправності.

Перевірте швидкість наростання імпульсів перемикачів, що відображається у показаннях для приводу. Час або швидкість наростання імпульсів вказується у вигляді значень  $dV/dt$  (швидкість зміни напруги з часом), які необхідно порівняти із зазначеною ізоляцією електродвигуна.

Перевірте частоту перемикачів фази постійного струму. Перевірте наявність потенційних несправностей електронного перемикача або заземлення – про ці несправності може свідчити сигнал, що плаває вгору та вниз.

Виміряйте асиметрію напруги, бажано при повному навантаженні. Асиметрія має перевищувати 2 %. Асиметрія напруги призводить до асиметрії струму, яка може призвести до надмірного нагрівання обмотки електродвигуна. Однією з причин виникнення асиметрії може бути несправність ланцюгів приводу. Несправність однієї з фаз називається «обривом фази», внаслідок якого електродвигун може нагріватися, не запускатися після зупинки, крім того, це може призвести до значного зниження ефективності, а також пошкодження електродвигуна та підключеного навантаження.

Виміряйте асиметрію струмів: вона повинна перевищувати 10 % для трифазних електродвигунів. Велика асиметрія при низькій напрузі може вказувати на наявність короткого замикання на обмотках електродвигуна або короткого замикання фаз на землю. Велика асиметрія може також призвести до спрацьовування приводу, високих температур електродвигуна та обгорання обмоток.

#### 4. Вхід електродвигуна

Напруга, подане на вхідні клеми електродвигуна, є ключовим показником, крім того, велике значення має вибір кабелю, що з'єднує привід з електродвигуном. Неправильний вибір кабелів може призвести до пошкодження приводу та електродвигуна через надмірну відбиту пікову напругу. Ці перевірки значною мірою аналогічні перевіркам на виході приводу, описаним вище. Перевірки

Переконайтеся, що струм на клеммах знаходиться в межах номіналу електродвигуна. Перевищення струму може спричинити нагрівання електродвигуна та знизити термін служби ізоляції статора, що може призвести до передчасної відмови електродвигуна.

Модуляція напруги допомагає визначити високі вершини напруги на землю, які можуть пошкодити ізоляцію електродвигуна.

Асиметрія струму, яка може значно вплинути термін служби електродвигуна і може бути ознакою наявності несправності перетворювача. Це може призвести до переривання напруги та спричинити спрацювання системи захисту від перевантаження.

Асиметрія струму може вказувати на асиметрію напруги або несправності випрямляча приводу.

#### 5. Напруга на кінцях валу електродвигуна

Імпульси напруги електроприводу можуть замикатися від статора електродвигуна до ротора, що призводить до появи напруги на валу ротора. Коли напруга на валу ротора перевищує ізоляційну здатність мастила підшипника, можуть виникнути струми іскрового розряду (іскріння), що призведе до утворення піттингової корозії та канавок на ободі підшипника електродвигуна, через що електродвигун може передчасно вийти з ладу. Перевірка

Виміряйте напругу між «масою» електродвигуна та валом приводу. Наприклад, модель MDA-550 для цієї мети оснащена датчиком із щіткою із

Апаратне забезпечення та система управління для триколісного транспортного засобу вуглецевого волокна. Ця перевірка дозволяє легко виявити наявність руйнівних струмів іскрового розряду, у той час як показання амплітуди імпульсу та лічильник подій дозволяють взяти необхідних заходів до виникнення несправності.

Після вдалої перевірки електродвигула та інвертора можна перевіряти працездатність програмного забезпечення.

#### **4.1.2 Тестування апаратно-програмної частини комплексу**

Програмна частина комплексу створена завдяки мові Python який можна протестувати за допомогою вбудованого фреймворка unittest, в якості демонстрації модульного тестування була створена проста програма яка повністю демонструє принцип роботи тестування. Ми будемо використовувати процес Test-Driven Development, розпочавши з тестів, а потім реалізувавши функціональність, щоб тести пройшли.

Незважаючи на те, що це хороша практика для розробки нашої програми Python у віртуальному середовищі, для цього прикладу це не буде обов'язковим, оскільки unittest поставляється з дистрибутивом Python, і нам не знадобляться жодні інші зовнішні пакети для створення нашого калькулятора.

Наш калькулятор буде виконувати прості операції складання, віднімання, множення та поділу між двома цілими числами. Ці вимоги будуть керувати нашими функціональними тестами з використанням фреймворку unittest.

Ми будемо тестувати чотири операції, що підтримуються нашим калькулятором, окремо та писати тести для кожної з них в окремому наборі тестів, оскільки тести конкретної операції, як очікується, будуть виконуватись разом. Наші тестові набори будуть розміщені в одному файлі, а наш калькулятор – в окремому.

Наш калькулятор буде простим калькулятором класу з функціями для обробки чотирьох операцій, очікуваних від нього. Давайте почнемо

Апаратне забезпечення та система управління для триколісного транспортного засобу  
тестування з написання тестів для операції додавання в нашому  
test\_simple\_calculator.py

```
import unittest
from simple_calculator import SimpleCalculator

class AdditionTestSuite(unittest.TestCase):
    def setUp(self):
        """ Executed before every test case """
        self.calculator = SimpleCalculator()

    def tearDown(self):
        """ Executed after every test case """
        print("\ntearDown executing after the test")

    def test_addition_two_integers(self):
        result = self.calculator.sum(5, 6)
        self.assertEqual(result, 11)

    def test_addition_integer_string(self):
        result = self.calculator.sum(5, "6")
        self.assertEqual(result, "ERROR")
```

Рисунок 4.3 – unites

Ми починаємо з імпорту модуля unittest і створення набору тестів (Addition TestSuite) для операції додавання.

У ньому ми створюємо метод setUp(), який викликається перед кожним тестовим випадком для створення нашого Простого калькулятора об'єкта, який буде використовуватись для виконання обчислень.

Метод tearDown() виконується після кожного тестового випадку, і оскільки в даний момент він нам не дуже потрібен, ми будемо використовувати його для роздрукування результатів кожного тесту.

Функції test\_addition\_two\_integers() , test\_addition\_integer\_string() та test\_addition\_negative\_integers() є нашими тестовими випадками. Передбачається, що калькулятор складе два позитивні або негативні цілі числа і поверне суму. При поданні цілого числа та рядка наш калькулятор повинен повертати помилку.

assertEqual() та assertNotEqual() – це функції, які використовуються для перевірки вихідних даних нашого калькулятора. Функція assertEquals()

Апаратне забезпечення та система управління для триколісного транспортного засобу перевіряє, чи рівні два наданих значення, у нашому випадку ми очікуємо, що сума 5 і 6 бути 11, тому ми порівнюємо це значення зі значенням, що повертається нашим калькулятором.

Якщо ці два значення рівні, тест пройдено. Інші функції затвердження, пропоновані unittest, включають:

`assertTrue(a)` : Перевіряє, чи наданий вираз є істинним

`assertGreater(a, b)` : Перевіряє, чи є a більше b

`assertNotIn(a, b)` : Перевіряє, чи знаходиться a в b

`assertLessEqual(a, b)` : Перевіряє, чи є a менше або одно b

Коли ми виконуємо тестовий файл, результат приведений на рис. 4.4:

```
$ python3 test_simple_calculator.py

tearDown executing after the test case. Result:
E
tearDown executing after the test case. Result:
E
tearDown executing after the test case. Result:
E
=====
ERROR: test_addition_integer_string (__main__.AdditionIntegerString)
-----
Traceback (most recent call last):
  File "test_simple_calculator.py", line 22, in test_addition_integer_string
    result = self.calculator.sum(5, "6")
AttributeError: 'SimpleCalculator' object has no attribute 'sum'
=====
ERROR: test_addition_negative_integers (__main__.AdditionNegativeIntegers)
-----
Traceback (most recent call last):
  File "test_simple_calculator.py", line 26, in test_addition_negative_integers
    result = self.calculator.sum(-5, -6)
AttributeError: 'SimpleCalculator' object has no attribute 'sum'
=====
ERROR: test_addition_two_integers (__main__.AdditionTwoIntegers)
-----
Traceback (most recent call last):
  File "test_simple_calculator.py", line 18, in test_addition_two_integers
    result = self.calculator.sum(5, 6)
AttributeError: 'SimpleCalculator' object has no attribute 'sum'
-----
Ran 3 tests in 0.001s
FAILED (errors=3)
```

Рисунок 4.4 – unittest результат 1



У верхній частині виводу ми бачимо виконання функції `tearDown()` через друк зазначеного нами повідомлення. Далі слідує буква `E` та повідомлення про помилку, що виникають при виконанні наших тестів.

Існує три можливі результати тесту: він може пройти, зазнати невдачі або зіткнутися з помилкою. Фреймворк `unittest` вказує три сценарії за допомогою:

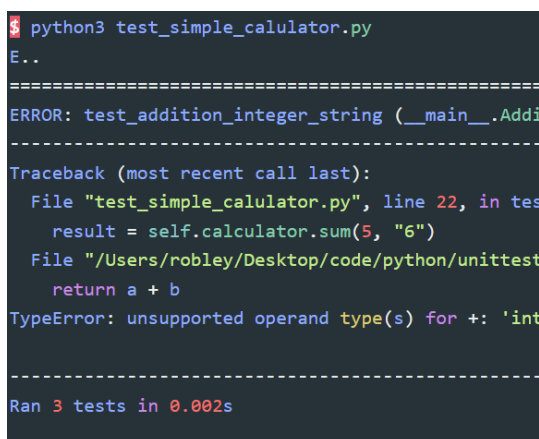
Повна зупинка (`.`): Показує проходження тесту

Літера `"F"`: Вказує на невдалий тест

Літера `"E"`: Вказує на помилку, що виникла під час виконання тесту

У нашому випадку ми бачимо букву `E`, що означає, що наші тести зіткнулися з помилками, які виникли під час виконання наших тестів. Ми отримуємо помилки, тому що ми ще не реалізували доповнення функціональності нашого калькулятора.

Наш калькулятор тепер готовий додати два числа, але щоб бути впевненим, що він буде працювати так, як очікувалося, давайте видалимо функцію `tearDown()` з наших тестів та запустимо наші тести ще раз рис. 4.5.



```
python3 test_simple_calculator.py
E..
=====
ERROR: test_addition_integer_string (__main__.AdditionTest)
-----
Traceback (most recent call last):
  File "test_simple_calculator.py", line 22, in test_addition_integer_string
    result = self.calculator.sum(5, "6")
  File "/Users/robley/Desktop/code/python/unittest/test_calculator.py", line 10, in sum
    return a + b
TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'
=====
Ran 3 tests in 0.002s
```

Рисунок 4.5 – unites результат 2

На цей раз наша функція `sum` виконується до завершення, але наш тест зазнає невдачі. Це відбувається тому, що ми не повертали жодного значення, коли один із вхідних даних не є цілим числом. Наше твердження порівнює `None` з `ERROR` і оскільки вони не рівні, тест не виконується. Щоб зробити наш тест прохідним, ми повинні повернути помилку до нашої функції `sum()`:

```
def sum(self, a, b):  
    if isinstance(a, int) and isinstance(b, int):  
        return a + b  
    else:  
        return "ERROR"
```

Рисунок 4.6 – відредагована частина коду

Всі наші тести проходять зараз, і ми отримуємо 3 повні зупинки, щоб вказати, що всі наші 3 тести для функціональності додавання проходять. Набори тестів на віднімання, множення та розподіл також реалізуються аналогічним чином.

Ми також можемо перевірити, чи є виняток. Наприклад, коли число ділиться на нуль, виникає виняток `ZeroDivisionError`. У нашому `Division TestSuite` ми можемо підтвердити, чи був викликаний виняток.

Функція `test_divide_by_zero_exception()` виконає функцію `divide(10, 0)` нашого калькулятора та підтвердить, що виняток дійсно був викликаний.

### 4.1.3 Тестування акумуляторної батареї пристрою

Основне завдання при перевірці стану будь-якої акумуляторної батареї - з'ясувати, чи володіє вона достатньою ємністю, чи може забезпечити заявлені виробником характеристики протягом необхідного часу. Проте безпосередньо засобами виміру визначаються лише кілька основних параметрів – напруга, сила струму. В акумуляторах, що обслуговуються, можна також заміряти щільність електроліту. Вимірювання можна проводити неодноразово, фіксуючи зміну значень з часом. Всі інші параметри та характеристики не вимірюються безпосередньо, а виводяться за розробленою виробником методикою, причому вона залежить від типу АКБ, і від рекомендацій виробника, і від виду підключеного навантаження. При цьому необхідно враховувати, що багато залежностей, що характеризують роботу АКБ, мають нелінійний характер. Можуть позначатися інші чинники, наприклад, вплив температури.

#### 1. Підключення навантаження

Апаратне забезпечення та система управління для триколісного транспортного засобу

До АКБ на деякий час підключається робоче або другорядне навантаження тієї чи іншої величини. Вольтметр або мультиметр вимірюється падіння напруги. Якщо процедура виконується кілька разів, між вимірами чекає певний час, щоб батарея відновилася. Отримані дані порівнюються з параметрами, заявленими виробником АКБ для даного типу батареї та даної величини навантаження.

## 2. Вимірювання за допомогою навантажувальної вилки

Будова найпростішої навантажувальної вилки показана на схемі рис.

4.7.

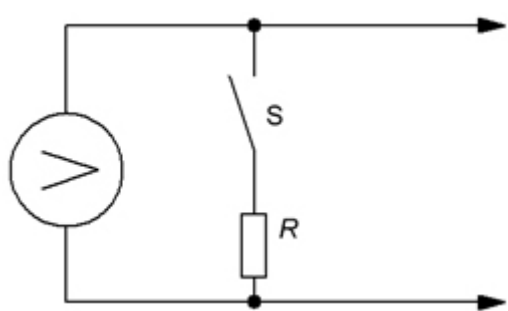


Рисунок 4.7 – Навантажувальна вилка

Пристрій оснащений вольтметром, паралельно якому встановлено великий за потужністю навантажувальний резистор, і має два щупи. У старих моделях аналогові вольтметри; нові моделі, як правило, оснащені РК-дисплеєм та цифровим вольтметром. Існують вилки навантаження з ускладненою схемою, що використовують кілька навантажувальних спіралей (змінних опорів), розраховані на різні діапазони вимірювання напруг, призначені для тестування кислотних або лужних акумуляторів. Є навіть вилки, якими тестують окремі банки акумуляторів. До складу просунутих пристроїв, крім вольтметра, може входити амперметр. Отримані при вимірах дані також необхідно зіставляти з параметрами, заявленими виробниками для типу батарей і цього опору.

## 3. Вимірювання за допомогою спеціальних пристроїв, тестерів аналізаторів АКБ

### Прилади Кулон

Принциповим розвитком ідеї навантажувальної вилки можна вважати сімейство цифрових приладів-тестерів Кулон (Кулон-12/6f, Кулон-12m, Кулон-12n та інші) для перевірки стану свинцевих кислотних акумуляторів, а також інші подібні пристрої. Вони дозволяють проводити швидкі виміри напруги, приблизно визначати ємність АКБ без контрольного розряду і зберігати в пам'яті кілька сотень, а іноді й тисяч вимірювань.

Прилади Кулон живляться від акумулятора, де проводяться вимірювання. Вхідні в комплект дроти з роз'ємами «крокодил» мають частини ізольовані один від одного, що забезпечує чотиризаписне підключення до акумулятора і усуває вплив на показання приладу опору в точках підключення затискачів. За заявою розробника, прилад аналізує відгук акумулятора на тестовий сигнал спеціальної форми, при цьому параметр, що вимірюється, приблизно пропорційний площі активної поверхні пластин акумулятора і, таким чином, характеризує його ємність. Власне, точність показань залежить від достовірності методики, розробленої виробником.

Ємність акумулятора - електричний заряд, що віддається повністю зарядженим акумулятором - вимірюється в ампер-годинниках і є витвір струму розряду на час. Для точного визначення ємності необхідно зробити розряд батареї (процес тривалий, багатогодинний), постійно фіксуючи величину заряду батареї. У цьому відносна ємність АКБ залежно від часу змінюється нелінійно. Наприклад, для акумуляторної батареї типу LCL-12V33AP відносна ємність змінюється з часом таблиця 4.1.

| Время разряда, часы | Относительная емкость, % |
|---------------------|--------------------------|
| 0,1                 | 37                       |
| 1,3                 | 48                       |
| 0,7                 | 53                       |
| 1,9                 | 76                       |
| 4,2                 | 84                       |
| 9,2                 | 92                       |
| 20                  | 100                      |

Таблиця 4.1 відносна ємність акумуляторної батареї

Прилад Кулон за допомогою швидкого вимірювання визначає ємність повністю зарядженого акумулятора. Він не призначений для оцінки ступеня зарядженості АКБ, всі вимірювання необхідно проводити повністю зарядженої батареї. Пристрій короткочасно подає тестовий сигнал, реєструє відгук від батареї і за кілька секунд видає орієнтовну ємність АКБ в ампер-годинниках. Одночасно на екран виводиться вимірювана напруга. Отримані значення можна зберігати у пам'яті приладу.

#### 4. Повна розрядка/зарядження

На сьогоднішній день повна розрядка та зарядка – це єдиний прямий та максимально достовірний спосіб визначення ємності АКБ. Спеціалізовані пристрої контролю розряду/заряду батареї (УКРЗ) дозволяють виконати глибоку розрядку та подальше повне зарядження батареї з постійним контролем ємності. Однак ця процедура займає дуже багато часу: 15-17-20-24 години, іноді і більше доби, залежно від ємності та поточного стану батареї. Хоча метод дає найточніші результати, через тимчасові витрати його застосування обмежене.

#### 4.1.4 Тестування сервера

Для тестування сервера застосуємо стрес тест.

Стрес-тестування зручно використовувати і для хмарних, і для фізичних серверів. Якщо ви працюєте у хмарах, ви зможете зрозуміти реальну пропускну спроможність вузла. Якщо розгортаєте систему на фізичному залозі - зможете переконатися, що із залізом все гаразд.

Стрес-тест дозволяє згенерувати навантаження на всі ключові системи сервера. Це дозволяє переконатися, що всі компоненти працюють нормально.

Інструментів для стрес тестування є ціла купа, ми зупинилися на `<var>stress-ng </var>`.

Стрес-тест – це внутрішній тест. На відміну від інструментів навантажувального тестування, наприклад, `<var><a href="https://ruhighload.com/Тестування+навантаження+на+сайт"`

Апаратне забезпечення та система управління для триколісного транспортного засобу  
`target="_blank" rel="nofollow">ab</a></ var>`, стрес-тести не полягають у генерації запитів до зовнішніх сервісів.

Стрес-тест дозволяє згенерувати навантаження на всі ключові системи сервера. Це дозволяє переконатися, що всі компоненти працюють нормально.

### Інструменти для стрес-тестів

Інструментів для стрес тестування є ціла купа, ми зупинилися на `<var>stress-ng </var>`.

Зверніть увагу, що стрес-тест – це внутрішній тест. На відміну від інструментів навантажувального тестування, наприклад, `<var><a href="https://ruhighload.com/Тестування+навантаження+на+сайт"`

`target="_blank" rel="nofollow">ab</a></ var>`, стрес-тести не полягають у генерації запитів до зовнішніх сервісів.

Статистика буде включати назву тесту та цифри за швидкістю виконання операцій. Абсолютні величини особливого значення немає. Проте їх варто порівнювати із цифрами серверів аналогічної конфігурації. Особливо цифри у колонках `<var>bogo ops/s</var>`.

### Тестування оперативної пам'яті

Для оперативної пам'яті є група тестів, куди входять операції виділення, копіювання та очищення пам'яті. Крім цього, цей набір містить деякі тести з класу `<var>cpu</var>`. Наприклад, тест компресії та сортування.

```
stress-ng --class memory --sequential 8 --timeout 60s --metrics-brief
```

група тестів для оперативної пам'яті

Під час цього тесту спостерігатимемо іншу картинку — велика кількість оперативної пам'яті (і своєї) буде зайнята:

The screenshot shows a terminal window with two sections. The top section displays system performance metrics for CPU, memory, and swap, along with task and load averages. The bottom section shows a list of running processes with columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command.

```

1  [|||||] 19.8% 5  [|||||] 16.4%
2  [|||||] 28.3% 6  [|||||] 12.3%
3  [|||||] 19.8% 7  [|||||] 15.5%
4  [|||||] 16.4% 8  [|||||] 12.2%
Mem [|||||] 31776/32804MB
Swap [|||||] 9594/16367MB
Tasks: 69, 31 thr; 1 running
Load average: 9.66 8.00 4.47
Uptime: 1 day, 00:43:27

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
22481 root 20 0 24892 484 468 S 0.0 0.0 0:00.32 -bash
26417 root 20 0 24284 388 384 S 0.0 0.0 0:00.08 -bash
17152 ubuntu 20 0 13652 364 360 S 0.0 0.0 0:00.08 /bin/bash /opt/vertica/agent/agent.sh /opt/vert
548 root 20 0 43452 412 408 S 0.0 0.0 0:00.18 /lib/systemd/systemd-logind
418 root 20 0 51624 772 652 S 0.0 0.0 0:00.23 /lib/systemd/systemd-udevd --daemon
F1|help F2|Setup F3|Search F4|Filter F5|Tree F6|SortBy F7|Nice F8|Nice F9|Kill F10|Quit

```

Рисунок 4.8 – Тестування пам'яті.

### Тестування дисків

Тут є дві групи тестів, які варто здійснити. Спочатку – група тестування низького рівня I/O пристроїв:

```
stress-ng --class io --sequential 8 --timeout 60s --metrics-brief
```

Ця група тестів включає створення/видалення файлів, запис блоків у файли та синхронізацію даних у файлах з диском.

Крім цього, має сенс запустити стрес-тест файлової системи. Він включає створення/видалення файлів та папок, переміщення по дереву файлів, створення посилань, блокування, перейменування тощо.

```
stress-ng --class filesystem --sequential 8 --timeout 60s --metrics-brief
```

Під час тестування дискової підсистеми можна спостерігати навантаження на диск за допомогою утиліти `iostat`:

```
iostat 5 -y
```

# Покаже статистику роботи дисків за останні 5 секунд

TL; DR

Використовуйте інструменти стрес-тестування, щоб перевірити роботу заліза та зрозуміти його реальну продуктивність. `stress-ng` – зручний інструмент для тестування серверів на Ubuntu. Якщо потрібно просто виконати всі тести, використовуйте команду:

```
iostat 5 -y
```

# Виконає більше 40 різних стрес-тестів сервера

Апаратне забезпечення та система управління для триколісного транспортного засобу  
Під час виконання тестів комп'ютер краще не чіпати. Після виконання статистика буде виведена у консолі.

#### **4.1.5 Тестування гальмівної системи**

Оскільки гальмівна система реалізована з використанням дискового тормозу лише на передньому колесі ефективність системи на високому рівні, але розглянемо параметри і норми яким повинна відповідати гальмівна система.

При роботі гальм енергія обертання перетворюється на теплову. Отже, гальмівний ефект створюється завдяки тертю. Воно відбувається між гальмівними накладками та барабанами (гальмівними дисками). Перед початком кожної подорожі необхідно обов'язково перевіряти стан гальмівної системи. Але навіть якщо ви ретельно відрегулювали обидва гальма, це не означає, що гальмівний ефект буде відповідним. Пояснюється це тим, що накладки зносилися нерівномірно або заклепки беруть участь у терті. Такий самий вплив надає і попадання масел на гальмівні колодки. Пошкоджені, односторонньо зношені накладки необхідно замінити, але при цьому необхідно з'ясувати причину одностороннього зношування та усунути дефект. Відшліфовані гальмівні накладки можна знову зробити шорсткими (знежиривши їх або втративши дрібнозернистою шкіркою). Однак якщо при цьому помітні заклепки, гальмівні накладки необхідно замінити. Вода, що потрапила на гальмівні колодки, може стати причиною прослизання, але після декількох різких гальмування вона випарується і несправність буде усунена.

Згідно з нормами (стандарт якості ISO 9001) гальмівний глях при початковій швидкості 100 км в годину не повинен перевищувати 40 метрів. Величина гальмівного шляху залежить від швидкості руху, стану покриття дороги, від справності гальм та інших чинників. Наприклад, при швидкості легкового автомобіля 30 км/год при різкому гальмуванні автомобіль проходить гальмівний шлях, що дорівнює 10 м. При швидкості 60 км/год



Апаратне забезпечення та система управління для триколісного транспортного засобу гальмівний шлях становитиме вже 40 м. Тобто при збільшенні швидкості вдвічі гальмівний шлях збільшується вчетверо. Гальмівний шлях набагато збільшується, якщо гальмування автомобіля відбувається на слизькій дорозі (дощ або сніг).

#### **4.2 Інструкція з використання Для використання**

Для використання програмно апаратного комплексу необхідно виконати необхідні установи для його коректного та безпечного використання

Для початку необхідно впевнитися що всі кріплення знаходяться в відповідному стані, а саме всі болтові з'єднання повинні пружно зажаті.

Перед початком руху необхідно впевнитися що акумуляторна батарея має відповідний рівень заряду та надійно закріплена на апаратному комплексі.

Необхідно перевірити гальмівну систему всі троси та ручки гальм повинні бути в справному стані, гальмівний диск не повинен бути забруднений.

Перед початком руху необхідно забезпечити себе та пасажирів обладнанням персонального захисту, а саме захисними шоломами.

В процесі пересуванні намагатися не перевантажувати силовий агрегат під час крутих підйомів, під час крутих схилів використовувати режим рекуперації для реалізації підзарядження акумулятора, та уникати використання тормозу впродовж великої кількості часу за для уникання виходу їх з ладу. Головне під час експлуатації триколісної моделі розуміння того що не завжди власні побажання та амбіції відповідають реаліям дійсності.

#### **Висновки до розділу 4**

В даному розділі розглянуто тестування програмної та апаратної складової триколісної системи. В ході роботи було розглянуто методи тестування окремих складових та приведені результати роботи тестування.

Було приведенно стислу інструкцію з використання триколісної моделі.

## **ВИСНОВКИ**

Метою бакалаврської роботи є розробка пристрою апаратного забезпечення та системи управління для триколісного засобу. Проведено аналіз характеристик та особливостей прототипів та вже існуючих комплексів які дають змогу пересуватися в просторі за рахунок перетворення електричної енергії в механічну, було обґрунтовано кожен систему та шляхи вдосконалення вже існуючих комплексів, серед наявних недоліків найбільш явним був рівень безпеки який надають комплекси бюджетного сегменту. При проектуванні апаратно-програмно комплексу передбачено реалізацію декількох режимів роботи апарату в частості режим ручного та автоматизованого управління за допомогою мобільного додатку. Також, при розробці схеми апаратної частини програмно-апаратного комплексу передбачено встановлення та модернізацію системи шляхом встановлення нових необхідних в роботі датчиків.

Програмне забезпечення для мікроконтролера написано на мові Python. Використання даної мови програмування дозволило максимально ефективно задіяти апаратні ресурси вибраного мікроконтролера, отримавши при цьому досить компактний та швидкий код.

Створений мобільний додаток має змогу передавати відео сигнал з апаратного комплексу та відсилати керуючі сигнали для пересування комплексу в просторі.

Після проектування програмної та апаратної частин пристрою його працездатність перевірено на розробленому макеті.

Серед подальших перспектив слід виділити детальне дослідження та реалізацію різних режимів роботи апарату що дозволить покращити характеристики та позбавитись від недоліків.

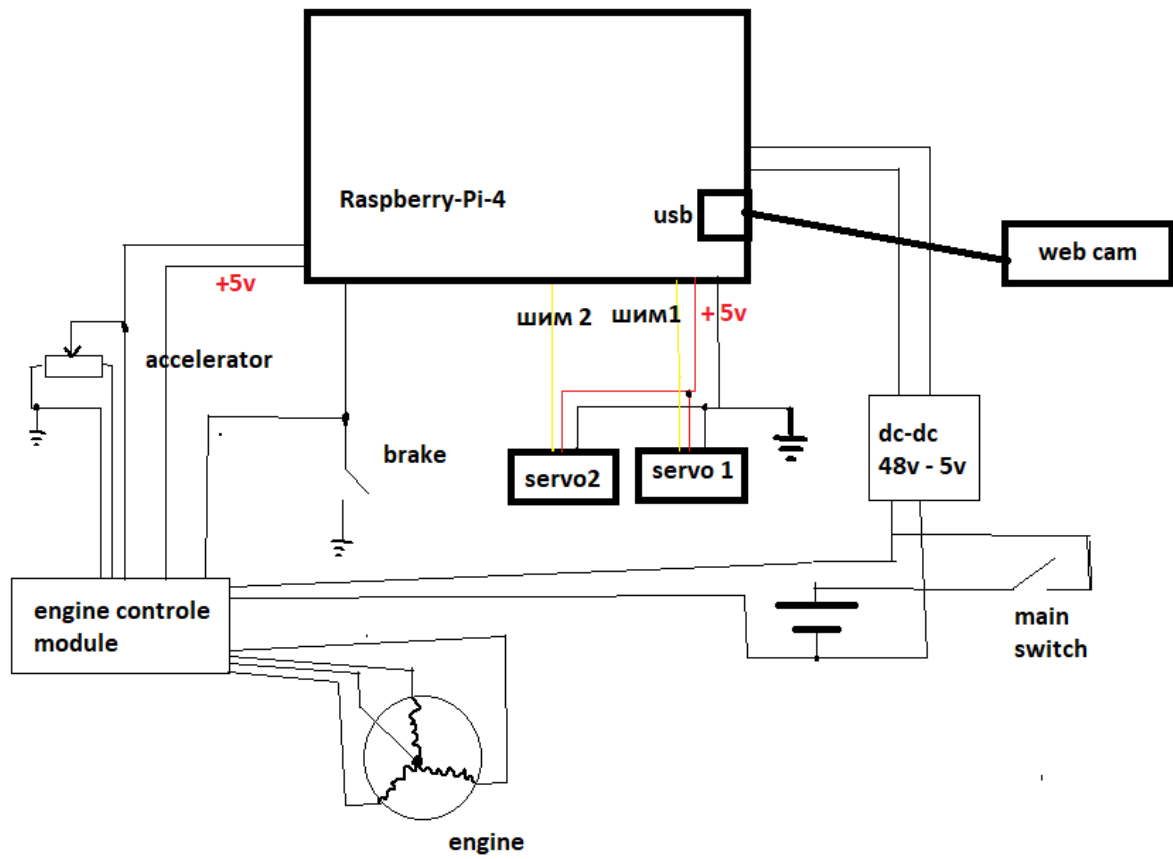
**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

- [1] Гироскутеры, гироборд, сигвей. [Онлайновий]. Available: <https://giroking.com.ua/>
- [2] Мощный недорогой электровелосипед [Онлайновий]. Available: <https://habr.com/ru/post/481208/>
- [3] Контроллеры для мотор колес [Онлайновий]. Available: <https://moto-velo.com.ua/kontrollery-dlja-motor-koles-i-jelektrodvigatelej-bldc.html>
- [4] Мотор колесо [Онлайновий]. Available: <https://electropowerbikes.com/motor-koleso/>
- [5] Сервоприводы: устройство [Онлайновий]. Available: <http://wiki.amperka.ru/articles:servo>
- [6] Сборка Аккумуляторной батареи на 48 вольт [Онлайновий]. Available: <https://elektro-velik.ru/elektrovelosiped/kak-sobrat-akkumulyator-elektrovelosipeda>
- [7] Бесплатный VPS для Forex [Онлайновий]. Available: [https://govpsfx.com/?utm\\_source=yandex&utm\\_medium=cpc&utm\\_term=vps\\_server\\_dlya\\_foreks&utm\\_content=besplatnye\\_vps\\_servera\\_fx&utm\\_campaign=reklama\\_rsy&type=search&source=none&block=premium&position=3&keyword=---autotargeting&yclid=3402948703449537830](https://govpsfx.com/?utm_source=yandex&utm_medium=cpc&utm_term=vps_server_dlya_foreks&utm_content=besplatnye_vps_servera_fx&utm_campaign=reklama_rsy&type=search&source=none&block=premium&position=3&keyword=---autotargeting&yclid=3402948703449537830)
- [8] Open Server Panel [Онлайновий]. Available: <https://ospanel.io/>
- [9] SMS-контроллер на базе SIM800L [Онлайновий]. Available: <https://sdelaysam--svoimirukami-ru.turbopages.org/s/sdelaysam-svoimirukami.ru/3652-sms-kontroller-na-baze-sim800l.html>
- [10] Servo Motor Manual [Онлайновий]. Available: <https://www.electronicwings.com/sensors-modules/servo-motor>
- [11] GSM и GPRS модули [Онлайновий]. Available: <https://arduinomaster.ru/datchiki-arduino/gsm-gprs-modul-arduino/>

- [12] Акселерометр. Виды и типы. Работа и применение. Особенности [Онлайновый]. Available: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/akselerometr/>.
- [13] Python Online Compiler [Онлайновый]. Available: <https://repl.it/languages/python3>.
- [14] Use Orange Pi by Using SSH and VNC Server [Онлайновый]. Available: <https://www.instructables.com/id/Use-Orange-Pi-Without-Monitor-by-Using-SSH-and-VNC/>.
- [15] Развертывание многопользовательской WebRTC трансляции [Онлайновый]. Available: <https://habr.com/ru/post/323176/>
- [16] Веб-сервер на Orange Pi PC» [Онлайновый]. Available: <https://orangepi.su/content.php?p=96&c=Veb-servernaOrangePiPS>
- [17] Пишем своё первое приложение на Android [Онлайновый]. Available: <https://habr.com/ru/post/109944/>
- [18] HTTPD - Apache2 интернет сервер [Онлайновый]. Available: [https://help.ubuntu.ru/wiki/руководство\\_по\\_ubuntu\\_server/web\\_сервера/httpd\\_apache2\\_web\\_server](https://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/web_сервера/httpd_apache2_web_server)
- [19] Установка и запуск веб-сервера Apache на Orange Pi Zero [Онлайновый]. Available: [http://hamlab.net/mcu/orangepi/orangepi\\_zero\\_apache.html](http://hamlab.net/mcu/orangepi/orangepi_zero_apache.html)
- [20] Real-time communication for the web [Онлайновый]. Available: <https://webrtc.org/>
- [21] WebRTC Media & Broadcasting Server [Онлайновый]. Available: <https://flashphoner.com/webrtc-server-potokovogo-video-dlya-onlajnt/?lang=ru>
- [22] Сервер ретрансляции потокового видео [Онлайновый]. Available: <https://www.linux.org.ru/forum/general/4324316>
- [23] Armbian [Онлайновый]. Available: <https://www.armbian.com/download/>

## Додаток А

### Принципова схема пристрою



**Додаток Б****Програмний код пристрою****Керування мотором**

```

#include "TimerThree.h" // good for 16kHz 1024 PWM
// PWM pins are 2,3,5
#include "OledI2Cbb.h" // BitBang I2C, because SDA 20 taken for Hall_W
#include "OledPix.h"
// defines
#define PAS 22 //Pedal Assist or 1:1 accelerator
#define Brake 23 // Low Brake signal
#define Throttle A0 // Analog Throttle signal
#define Hall_U 18 // BLUE Hall wire ( Interrupt pins !)
#define Hall_V 19 // YELLOW Hall wire
#define Hall_W 20 // GREEN Hall wire
#define Assist_minus 30 // push Button -
#define Assist_plus 31 // push Button +
#define H1 2 // active High PWM, BLUE motor wire
#define H2 3 // YELLOW motor wire
#define H3 5 // GREEN motor wire

#define L1 4 // active Low !!! BLUE
#define L2 6 // YELLOW
#define L3 7 // GREEN

// Globals
// Tick Generators
static uint32_t last_time, now = 0; // OLED every 1100 ms to avoid video flicker
volatile uint32_t last_time2, now2 = 0; // Detect Standstill after 200ms
static uint32_t last_time3, now3 = 0; // Push-button scan period 200ms
static uint32_t last_time4, now4 = 0; // RPM calculation and OLED every 1000 ms

volatile uint32_t Motor_tick,last_Motor_tick = 0; // to calculate speed, distance, Odometer

boolean toggle=LOW;
boolean Motor_ON,last_PAS,start_pulse =false;
volatile byte Hall,Hall_simu,last_Hall,motor_turns,PAS_ticks, assist_level=0;
unsigned int meter_per_hour, pw, pw_p, pw_t, pw_max, throttle_in =0;

void Read_Hall(){
    Hall=0;
    if (digitalRead(Hall_U)==HIGH) Hall=Hall+1;
    if (digitalRead(Hall_V)==HIGH) Hall=Hall+2;
    if (digitalRead(Hall_W)==HIGH) Hall=Hall+4;
}

```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
}  
void setup() {  
    // put your setup code here, to run once:  
  
    // 6 FET H-Bridge  
    digitalWrite(H1,LOW); // disable, active High  
    pinMode(H1,OUTPUT);  
    digitalWrite(H2,LOW);  
    pinMode(H2,OUTPUT);  
    digitalWrite(H3,LOW);  
    pinMode(H3,OUTPUT);  
  
    digitalWrite(L1,HIGH); // disable, active Low  
    pinMode(L1,OUTPUT);  
    digitalWrite(L2,HIGH);  
    pinMode(L2,OUTPUT);  
    digitalWrite(L3,HIGH);  
    pinMode(L3,OUTPUT);  
  
    Timer3.initialize(60); // initialize timer3 60us period  
    Timer3.pwm(H1,0);      // setup pwm on pin 2, 0 duty cycle  
    Timer3.pwm(H2,0);      // pin 3  
    Timer3.pwm(H3,0);      // pin 5,  
  
    pinMode(Hall_U,INPUT); // remove INPUT_PULLUP because external filter  
    pinMode(Hall_V,INPUT);  
    pinMode(Hall_W,INPUT);  
  
    Read_Hall();  
    last_Hall=Hall;  
  
    pinMode(13,OUTPUT);  
    digitalWrite(13,LOW); //shut down the LED  
  
    last_time=millis(); // init time-outs  
    last_time2=millis();  
    last_time3=millis();  
    last_time4=millis();  
  
    // Serial Monitor  
    Serial.begin(115200);  
  
    pinMode(PAS,INPUT_PULLUP);  
    pinMode(Brake,INPUT_PULLUP);  
    pinMode(Assist_minus,INPUT_PULLUP);
```

Апаратне забезпечення та система управління для триколісного транспортного засобу  
pinMode(Assist\_plus, INPUT\_PULLUP);

```
last_PAS=digitalRead(PAS);
```

```
// OLED init
```

```
SetupI2Cbb();
```

```
SetupOledPix();
```

```
// Motor driven by interrupts
```

```
attachInterrupt(digitalPinToInterrupt(Hall_U), next_step, CHANGE);
```

```
attachInterrupt(digitalPinToInterrupt(Hall_V), next_step, CHANGE);
```

```
attachInterrupt(digitalPinToInterrupt(Hall_W), next_step, CHANGE);
```

```
}
```

```
void loop() {
```

```
    // put your main code here, to run repeatedly:
```

```
// When Motor ON, detect standstill by time-out and give a pulse forward
```

```
if (Motor_ON==true){
```

```
    Read_Hall();
```

```
    if (start_pulse==true) {
```

```
        switch (Hall){
```

```
            case 1: Hall_simu=3;break;
```

```
            case 2: Hall_simu=6;break;
```

```
            case 3: Hall_simu=2;break;
```

```
            case 4: Hall_simu=5;break;
```

```
            case 5: Hall_simu=1;break;
```

```
            case 6: Hall_simu=4;break;
```

```
            default: Hall_simu=1;break;
```

```
        }
```

```
        Hall=Hall_simu;
```

```
        next_step();
```

```
    }
```

```
    now2=millis();
```

```
if (( now2-last_time2)>=200 ) // Timeout 200 ms to start Standstill
```

```
{
```

```
    last_time2=now2; // prepare for next loop
```

```
    start_pulse=true;
```

```
}
```

```
}
```

```
else // Motor OFF
```

```
{
```



Апаратне забезпечення та система управління для триколісного транспортного засобу

```

Timer3.setPwmDuty(H1, 0);
    Timer3.setPwmDuty(H2, 0);
    Timer3.setPwmDuty(H3, 0);
    digitalWrite(L1,HIGH);
    digitalWrite(L2,HIGH);
    digitalWrite(L3,HIGH);
    delayMicroseconds(100);
}

calc(); // All computations to calculate PWM ,other values and start/stop Logic

} // end of Loop

void calc(void){

// Set maximum Power according to Assistance Level (0...5)
//Convert assist_level to pw_max
    switch(assist_level){
        case 1: pw_max=100;break;
        case 2: pw_max=200;break;
        case 3: pw_max=400;break;
        case 4: pw_max=600;break;
        case 5: pw_max=1000;break;
        default:pw_max=0;break;
    }

// Read Throttle
throttle_in=analogRead(Throttle); // read throttle 0..1024
if (throttle_in>200) Motor_ON=true; // motor ON by throttle
if ((throttle_in<200)&(PAS_ticks==0)) Motor_ON=false; // motor OFF by throttle

// Count PAS_ticks and Motor ON with PAS
if (digitalRead(PAS)!=last_PAS) {
    last_PAS=digitalRead(PAS);
    PAS_ticks++;
    Motor_ON=true; // motor ON by PAS
}

if (digitalRead(Brake)==LOW) {
    Motor_ON=false;// Brake stops motor
    throttle_in=0;
}

// pw calculation, a fraction of pw_max, depending from PAS and Trottle
//throttle influence here

```

```

    Апаратне забезпечення та система управління для триколісного транспортного засобу
    pw=pw_p; // PAS influence
    pw_t=(pw_max*(throttle_in/32))/32; // calculation could be done by shifting
    if (pw_t>pw) pw=pw_t; // trottle influence

// check Push Button Inputs
now3=millis();
if (( now3-last_time3)>=200 ) // every 200ms, gives debouncing
{
    last_time3=now3;

    // set assist_level 0..5 by push button
    if (digitalRead(Assist_minus)==LOW) assist_level--;
    if (assist_level>128) assist_level=0; // min value
    if (digitalRead(Assist_plus)==LOW) assist_level++;
    if (assist_level>=6) assist_level=5; // max value

}

// every sec. calculations like rpm
now4=millis();
if (( now4-last_time4)>=1000 )
{
    last_time4=now4;

    meter_per_hour=((Motor_tick-last_Motor_tick)*223)/1000;
    last_Motor_tick=Motor_tick;

    // Motor OFF if PAS frequency=0
    if (Motor_ON==true) {
        if ((PAS_ticks==0)&(throttle_in<200)) Motor_ON=false;
    }
    // calculate assit with PAS
    if (PAS_ticks>10) PAS_ticks=10; // limit 10/sec.
    pw_p=(pw_max*PAS_ticks)/10; // PAS influence on pw

    //Serial.print("PAS/sec. : ");
    //Serial.println(PAS_ticks);
    PAS_ticks=0;

    Serial.print("speed meter km/h: ");
    Serial.println(meter_per_hour);

```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```

Serial.print("Motor : ");
Serial.println(Motor_ON);
Serial.print("Throttle : ");
Serial.println(throttle_in);
Serial.print("Assist Level (0-5) : ");
Serial.println(assist_level);
Serial.print("pw_max : ");
Serial.println(pw_max);
Serial.print("pw (0-1000) : ");
Serial.println(pw);
Serial.print("Hall_simu : ");
Serial.println(Hall_simu);
Serial.print("Hall : ");
Serial.println(Hall);

}

// every 1100ms OLED refresh
now=millis();
if (( now-last_time)>=1100 )
{
    last_time=now;

    // OLED Display with modified Didel Library
    LiCol(0,10);
    Text("    ArduBike 250W");

    LiCol(7,0);
    BigHex81(assist_level);

    LiCol(7,72);
    BigDec99(meter_per_hour);
}

}

```

## Веб сервер

## Steps

1. [Install Raspbian onto SD Card](#step-1)
2. [Boot from USB (optional)](#step-2)
3. [Initial Configuration](#step-3)
4. [Internal Networking](#step-4)
5. [Install all the things](#step-5)
6. [Get your code, Run your code](#step-6)
7. [Proxy requests with nginx](#step-7)

2022 p.

Чупріков М. К.

6.050102 – МР.ПЗ.00 – 605.21610611

Апаратне забезпечення та система управління для триколісного транспортного засобу

8. [Make your web server available from ... the web.](#step-8)
9. [Enable SSL like a boss (optional)](#step-9)
10. [Enable Cross Origin Requests (optional)](#step-10)
11. [Step 11: BONUS! Control GPIO pins with Node.js](#step-11)

## <a name="step-1"></a> Step 1: Install Raspbian onto SD Card

On your development machine, [Download Raspbian Lite](https://www.raspberrypi.org/downloads/raspbian/). Unzip that file to get the `.img``.

If your dev machine is Windows or Linux, you can lookup [directions online](https://www.raspberrypi.org/documentation/installation/installing-images/README.md).

If you are using a Mac:

Format SD card, MS-DOS (FAT) using Disk Utility. Run ``diskutil list`` from terminal to get number of SD Card disk.

Then to copy the OS to the SD card, run (changing the location and filename of the `.img` file you just downloaded):

...

```
sudo dd bs=1m if=~/.Downloads/2017-03-02-raspbian-jessie-lite.img
of=/dev/rdisk{NUMBER_OF_SD_CARD_DISK}
```

...

After a while, you should see output similar to

...

```
1329+0 records in
1329+0 records out
1393557504 bytes transferred in 367.987715 secs (3786967 bytes/sec)
```

...

Once done, eject the SD card by running.

...

```
sudo diskutil eject /dev/disk{NUMBER_OF_SD_CARD_DISK}
```

...

Now plug that SD card, a monitor and a keyboard into your Raspberry Pi, power it up and dive in.

*\*Note\** - By default Raspbian ships with the British keyboard layout, which makes some keys hard to find. If you can't find the ``|`` character, ensure Num Lock is on, hold alt, then press 1,2,4 on the keypad.

## <a name="step-2"></a>Step 2: Boot from USB (optional)

Апаратне забезпечення та система управління для триколісного транспортного засобу

If your paranoid about the write limits of SD cards over time, or have been burned by a corrupted SD card one too many times, you can instruct your Pi to boot from a USB flash drive instead. If you're not worried about this, skip ahead to [Step 3](#step-3).

If you have a Raspberry Pi 3, there are [new instructions](<https://www.raspberrypi.org/documentation/hardware/raspberrypi/bootmodes/msd.md>) on how to accomplish this. The cool thing about this, is that after following those steps, you can remove the SD and boot entirely from USB. Pretty cool.

Login using username `pi` and password `raspberry`, then:

```
...
```

```
sudo apt-get update && sudo apt-get upgrade
echo program_usb_boot_mode=1 | sudo tee -a /boot/config.txt
sudo reboot
```

```
...
```

Log back in as `pi`, and run the following command:

```
...
```

```
vcgencmd otp_dump | grep 17:
```

```
...
```

If that outputs `17:3020000a`, congrats, you win.

Now you need to add Raspbian to your USB drive. On your dev machine, repeat [Step 1](#step-1), only with the USB drive, not the SD card. Once that finishes up, shutdown the Raspberry Pi (`sudo shutdown -h now`), eject the SD card, insert your USB drive and plug her in.

If all went as planned your Pi should startup as normal after 5-10 seconds and you're now done with the SD card for good!

If you have a Pi older than 3, you can still run your server off of a USB stick, but you'll still need the SD card for booting up. Instructions for that can be [found here](<http://jonathanmh.com/boot-raspberry-pi-from-a-usb-stick/>).

## [Step 3](#)) Initial Configuration

Using `sudo raspi-config`, change hostname, enable SSH, wait for network on boot, reduce GPU to 16MB.

Optionally, you can edit the message that appears in the terminal when you first log in. I like to add some [ascii art](<http://patorjk.com/software/taag/>) to match the Pi's host name. To do so, run:

```
...
```

```
sudo nano /etc/motd
```

```
...
```

and paste in the message you want displayed.

```
### Setup new user
```

It's generally a good idea to have a different linux user for each app running on your server.

To create a new user, for an `api` app for example:

```
...
```

```
sudo useradd -m api -G sudo
```

```
sudo passwd api
```

```
logout
```

```
...
```

Login with the user and password you just created. After you've confirmed your new user has sudo privileges, you can delete the default `pi` user for good measure:

```
...
```

```
sudo deluser pi
```

```
...
```

```
## <a name="step-4"></a> Step 4) Internal Networking
```

You can plug your Pi via ethernet cable to your router or setup Wifi.

To connect to Wifi: `sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf`. Add the following:

```
...
```

```
network={
```

```
ssid="SSID"
```

```
psk="WIFI PASSWORD"
```

```
}
```

```
...
```

Most wireless routers these days are also DHCP servers, which means by default, you will get a different IP address every time you connect to the network. This isn't ideal, as you'll need to know where to find your server on the network if you're going to route requests to or SSH into it.

To assign Static IP address to your Pi on the internal network:

```
...
```

```
sudo nano /etc/dhcpd.conf
```

```
...
```

Add the following:

```
...
```

```
interface wlan0
```

```
# Desired IP Address. Keep /24
```

```
static ip_address=192.168.0.200/24
```

```
# IP Address of router
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
static routers=192.168.0.1
# IP Address of router
static domain_name_servers=192.168.0.1
...

```

Restart Wifi:

```
...
sudo ifdown wlan0
sudo ifup wlan0
...

```

If that is done correctly, you can now connect to your Pi via SSH on the local network. What that means is you can ditch the keyboard and monitor that are plugged into your Pi, and do the rest of the work from your normal dev machine. Woo!

From the terminal on your dev machine (or whatever SSH client you use on Windows), you can run

```
...
ssh api@192.168.0.200
...

```

Use the password you set when you created the new user.

## [Step 5](#)) Install all the things:

Pull in reference to Node v6 (latest LTS version). Change version digit in the URL if you want a different version ([https://deb.nodesource.com/setup\\_7.x](https://deb.nodesource.com/setup_7.x) for example)

```
...
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
...

```

Install the things

```
...
sudo apt-get update
sudo apt-get install git nodejs nginx -y
...

```

Install will take a decent amount of time to run (10-20 minutes)

Install yarn and pm2 with npm

```
...
sudo npm install -g yarn pm2
...

```

### Configure PM2 Logrotate (optional)

Апаратне забезпечення та система управління для триколісного транспортного засобу PM2, the process manager that keeps your Node scripts running, outputs great log files. However, without configuration these log files can eat up more and more storage on you Pi. The `pm2-logrotate` module can help with this issue.

```
...
# Install the module
pm2 install pm2-logrotate
# Keep at most 90 days of logs
pm2 set pm2-logrotate:retain 90
# Gzip old log files to save space
pm2 set pm2-logrotate:compress true
...
```

### ### Upgrade Nginx (optional)

Installing Nginx in the normal way like we just did will install a much older version of the software, something like 1.6. If you are going want to do modern things, including taking advantage of the HTTP/2 protocol, you're going to need a newer version of Nginx (at least 1.9.5). If you're fine with the older version, feel free to skip this section.

After installing nginx in the normal way above, remove it with `apt-get`:

```
...
sudo apt-get remove nginx
...
```

We are going to download and compile the newest version of Nginx, along with OpenSSL and PCRE. Luckily, a guy by the name of [Matt Wilcox]() has written [a script to do exactly that on a Raspi](). Download that script via wget:

```
...
cd ~
wget
https://gist.githubusercontent.com/MattWilcox/402e2e8aa2e1c132ee24/raw/1a4f1f78cb69b715d7e63a6c9b97032bc9ddeb8/build_nginx.sh
...
```

Check for the latest version numbers of [OpenSSL](<https://www.openssl.org/source/>) (version 1.0.2d), [PCRE](<http://www.pcre.org/>) (version 8.39) and [Nginx](<http://nginx.org/en/download.html>) (version 1.9.7). The latest versions I could get to work are in parenthesis.

Edit variables at the top of the build script we just downloaded to reflect these version numbers.

```
...
```



Апаратне забезпечення та система управління для триколісного транспортного засобу

```
sudo nano ~/build_nginx.sh
```

```
...
```

Make the file executable and run it:

```
...
```

```
sudo chmod +x build_nginx.sh
```

```
sudo ./build_nginx.sh
```

```
...
```

This will take a while to run. Be chill.

### Tweak Nginx (optional)

By default Nginx will advertise it's exact version number, in the `server` header and a couple other places. You can stop this behavior by editing the nginx config file:

```
...
```

```
sudo nano /etc/nginx/nginx.conf
```

```
...
```

and ensuring the following line is NOT commented out:

```
...
```

```
server_tokens off;
```

```
...
```

Nginx also ships with a default site that just shows a "Welcome to nginx on Debian!" page. This isn't really useful to us, and since we'll be adding new nginx sites, we can disable this default site. To do so, run:

```
...
```

```
sudo nano /etc/nginx/sites-available/default
```

```
...
```

and add `return 404;` in the line right after the listen directives.

Reload nginx to see our changes:

```
...
```

# Test your changes

```
sudo nginx -t
```

# If that says OK, reload

```
sudo service nginx restart
```

```
...
```

### MongoDB

If you want mongod v2.4, just run:

```
...
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
sudo apt-get install mongodb
```

```
...
```

If you want v3.0.14 (which is the highest version of Mongo that supports a 32bit system), [do the following](<http://andyfelong.com/2017/03/mongodb-3-0-14-binaries-for-raspberry-pi-3/>):

Check for mongodb user

```
...
```

```
grep mongodb /etc/passwd
```

```
# if NO mongodb user, create:
```

```
sudo adduser --ingroup nogroup --shell /etc/false --disabled-password --gecos "" \
--no-create-home mongodb
```

```
...
```

Download Binaries

```
...
```

```
sudo apt-get upgrade
```

```
mkdir ~/mongo
```

```
# Core Mongo
```

```
wget -P ~/mongo/ http://andyfelong.com/downloads/core_mongodb_3_0_14.tar.gz
```

```
# Mongo Tools
```

```
wget -P ~/mongo/ http://andyfelong.com/downloads/tools_mongodb_3_0_14.tar.gz
```

```
# Unzip Core files:
```

```
tar -xvzf core_mongodb_3_0_14.tar.gz
```

```
# Move the files where they need to go
```

```
sudo chown root:root mongo*
```

```
sudo chmod 755 mongo*
```

```
sudo strip mongo*
```

```
sudo cp -p mongo* /usr/bin
```

```
# Log and config directories
```

```
sudo mkdir /var/log/mongodb
```

```
sudo chown mongodb:nogroup /var/log/mongodb
```

```
sudo mkdir /var/lib/mongodb
```

```
sudo chown mongodb:root /var/lib/mongodb
```

```
sudo chmod 775 /var/lib/mongodb
```

```
cd /etc
```

```
sudo nano mongodb.conf
```

```
...
```

Add the following text to mongodb.conf:

```
...
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
bind_ip = 127.0.0.1
```

```
quiet = true
```

```
dbpath = /var/lib/mongodb
```

```
logpath = /var/log/mongodb/mongod.log
```

```
logappend = true
```

```
storageEngine = mmapv1
```

```
...
```

```
__The wiredTiger storage engine isn't supported on the 32-bit ARM chip. :'( __
```

```
Add to `service`:
```

```
...
```

```
cd /lib/systemd/system
```

```
sudo nano mongodb.service
```

```
...
```

```
Add the following text to `mongodb.service`:
```

```
...
```

```
[Unit]
```

```
Description=High-performance, schema-free document-oriented database
```

```
After=network.target
```

```
[Service]
```

```
User=mongodb
```

```
ExecStart=/usr/bin/mongod --quiet --config /etc/mongodb.conf
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
...
```

```
Add the following line to `~/.bashrc` (it's a setting that mongo needs to run):
```

```
...
```

```
export LC_ALL=C
```

```
...
```

```
Start up mongo!
```

```
...
```

```
sudo systemctl unmask mongodb
```

```
sudo service mongodb start
```

```
...
```

```
Enable all the mongo tools:
```

```
...
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
cd ~/mongo
mkdir tools
tar -xzvf tools_mongodb_3_0_14.tar.gz --directory tools/
cd tools
sudo strip ./*
sudo chown root:root ./*
sudo chmod 755 ./*
sudo mv ./* /usr/bin
...
```

Mongo 3 on your Raspi 3. Boomshakalaka.

## <a name="step-6"></a> Step 6) Get your code, Run your code

Generate SSH Key

...

```
ssh-keygen -f ~/.ssh/id_rsa -C "id_rsa"
```

...

In a browser, go to Github or Bitbucket and Add SSH Key OR Deploy/Access Key in your repository. Paste in output from:

...

```
cat ~/.ssh/id_rsa.pub
```

...

Clone your repository into your home dir.

...

```
cd ~ && git clone git@URL_OF_REPOSITORY.git
```

...

If needed, add app config env vars to .bashrc

...

```
sudo nano ~/.bashrc
```

```
# export APP_CONFIG_OPTION=foobar
```

...

Run your code with PM2:

...

```
pm2 start ~/APP_DIRECTORY/index.js
```

...

Make sure your app starts up when raspi restarts:

...

```
pm2 startup
```

```
## This command will tell you to run another command as sudo. Do that.
```

```
pm2 save
```

...

If you run into issues with your PM2 not starting on reboot, even after running the above commands, you can add the following cronjob: `crontab -e`

```
...  
@reboot cd ~/APP_DIRECTORY/ && pm2 start index.js  
...
```

Your app should now be available to any computer on your local network. You can visit

[192.168.0.200:3000](http://192.168.0.200:3000)

and you'll see your app running (where the IP is the static internal IP address you assigned the PI, and the port is port your Node.js code is listening on).

## [Step 7](#)) Proxy requests with nginx

Your node app most likely isn't running on port 80. It turns out it's kind of a pain to get node servers to directly listen on port 80. Enter nginx.

Add new nginx site block:

```
...  
sudo nano /etc/nginx/sites-available/{APP_NAME}  
sudo service nginx reload  
...
```

Add the following text:

```
...  
upstream {APP_NAME} {  
    server 127.0.0.1:{NODE_APP_PORT};  
    keepalive 64;  
}  
  
server {  
    listen 80;  
  
    server_name {DESIRED_DOMAIN_NAME};  
  
    location / {  
        proxy_pass http://{APP_NAME}/;  
        proxy_http_version 1.1;  
        proxy_cache_bypass $http_upgrade;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```

proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_max_temp_file_size 0;
proxy_redirect off;
proxy_read_timeout 240s;
}
}
...

```

Enable newly configured site:

```

...
sudo ln -s /etc/nginx/sites-available/{APP_NAME} /etc/nginx/sites-enabled/
sudo service nginx reload
...

```

You can edit the hosts file (/etc/hosts) on your dev machine to point your newly added domain name to your Raspberry Pi's static internal IP to test your nginx config.

## [Step 8](#)) Make your web server available from ... the web.

If you don't have a static IP from your internet provider (not sure? Odds are you don't), you'll need to use a Dynamic DNS service.

Head over to [Duck DNS](<http://www.duckdns.org/>). It's free and awesome. Make an account and claim a domain. Follow instructions on [this page](<https://www.duckdns.org/install.jsp?tab=linux-cron>).

```

...
mkdir ~/duckdns
cd ~/duckdns
nano duck.sh
...

```

Add the following line to `duck.sh`

```

...
echo url="https://www.duckdns.org/update?domains={YOUR_DOMAIN}&token={YOUR_TOKEN}4&ip=" | curl
-k -o ~/duckdns/duck.log -K -
...

```

Make the file executable:

```

...
chmod 700 duck.sh
...

```

Апаратне забезпечення та система управління для триколісного транспортного засобу  
Run that file every 5 minutes. Edit your crontab (`crontab -e`) and add the following line:

```
...
*/5 * * * * ~/duckdns/duck.sh >/dev/null 2>&1
...
```

Test the script:

```
...
./duck.sh
cat duck.log
...
```

If it's says `OK`, we good.

### Forward incoming requests to your network

You'll need to hop into your router's admin screen and forward incoming requests on specific ports to your raspi. [PortForward.com/](https://portforward.com/) has more info on specifics on how to do this on various routers.

Forward only ports 80 (HTTP), 443 (HTTPS), and 22 (SSH) to the static internal IP you assigned your Pi.

Once that's done correctly, you should be able to go to `{APP_NAME}.duckdns.org`([http://{APP\\_NAME}.duckdns.org](http://{APP_NAME}.duckdns.org)) and see your app. Woo.

#### Use your own domain

Own a domain name? If you've spent this much time hacking on a raspberry pi, I'm going to bet you do.

You can add a CNAME record to access your site via a submain of the domain name you own. The CNAME record should point to your `{APP_NAME}.duckdns.org` domain and the host should be your desired subdomain.

Once that change is saved and propogates, you should be able to access your Node.js app via a domain like:

```
[node-is-cool.myawesomedomain.com](http://node-is-cool.myawesomedomain.com)
```

### [Step 9\) Enable SSL like a boss \(optional\)](#)

Let's Encrypt makes enabling https free and relatively easy. You're already this deep in. Why stop now? [[Link](https://www.linuxbabe.com/linux-server/install-lets-encrypt-free-tlssl-certificate-nginx-debian-8-server)](https://www.linuxbabe.com/linux-server/install-lets-encrypt-free-tlssl-certificate-nginx-debian-8-server)

```
...
sudo nano /etc/apt/sources.list
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

...

and add the following line to the end:

...

```
deb http://ftp.debian.org/debian jessie-backports main
```

...

Install the client:

...

```
sudo apt-get update
sudo apt-get install letsencrypt -t jessie-backports
```

...

Get a cert:

...

```
sudo service nginx stop
sudo letsencrypt certonly --email <your-email-address> -d <your-domain-name>
```

...

Select option 2, standalone.

Renew the cert automatically by running:

...

```
mkdir ~/log
sudo crontab -e
```

...

and pasting:

...

```
# Check for cert renewal every Monday at 2:30am
30 2 * * 1 /usr/bin/certbot renew >> /home/{username}/log/le-renew.log
# Reload Nginx every Monday at 2:35am, to use potential new cert
35 2 * * 1 /etc/init.d/nginx reload
```

...

Now we need to configure nginx to serve your app over SSL.

Generate strong Diffie-Hellman key (this will take a while to run):

...

```
sudo mkdir /etc/nginx/ssl/
sudo openssl dhparam -out /etc/ssl/certs/dhparam2048.pem 2048
```



...

Add SSL params config file:

...

```
sudo nano /etc/nginx/snippets/ssl-params.conf
```

...

And paste in:

...

```
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
ssl_ecdh_curve secp384r1;
ssl_dhparam /etc/ssl/certs/dhparam2048.pem;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;" always;
add_header X-Frame-Options "SAMEORIGIN";
add_header X-Content-Type-Options nosniff;
add_header Referrer-Policy "no-referrer";
add_header X-XSS-Protection "1; mode=block";
```

...

Note: If you did not upgrade Nginx, remove the word `always` from the `Strict-Transport-Security` line, so it reads:

...

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;";
```

...

See [cipherli.st](https://cipherli.st/) for explanations for all of these SSL settings.

Edit your nginx site config file:

...

```
sudo nano /etc/nginx/site-available/{APP_NAME}
```

...

Апаратне забезпечення та система управління для триколісного транспортного засобу

Delete everything and paste in the following:

```
...
upstream {APP_NAME} {
    server 127.0.0.1:{NODE_APP_PORT};
    keepalive 64;
}

server {
    listen 80;
    server_name {DOMAIN_NAME};
    return 301 https://$host$request_uri;
}

server {

    listen 443 ssl http2;

    server_name {DOMAIN_NAME};

    ssl_certificate /etc/letsencrypt/live/{DOMAIN_NAME}/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/{DOMAIN_NAME}/privkey.pem;

    include snippets/ssl-params.conf;

    location / {
        proxy_pass http://{APP_NAME}/;
        proxy_http_version 1.1;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_max_temp_file_size 0;
        proxy_redirect off;
        proxy_read_timeout 240s;
    }
}
...
```

Note: If you did not upgrade Nginx, remove `http2` from the `listen 443` line so it reads:

```
...
listen 443 ssl;
...
```

Апаратне забезпечення та система управління для триколісного транспортного засобу  
Test your nginx config:

...

```
sudo nginx -t
```

...

If all good, reload nginx:

...

```
sudo service nginx reload
```

...

## [Step 10: Enable Cross Origin Requests \(optional\)](#)

If your app is an API that will be called via ajax on other domains, we'll need to adjust our nginx site config to tell browsers that we know what we're doing.

...

```
sudo nano /etc/nginx/site-available/{APP_NAME}
```

...

Delete everything and paste in the following:

...

```
upstream {APP_NAME} {
    server 127.0.0.1:{NODE_APP_PORT};
    keepalive 64;
}

server {
    listen 80;
    server_name {DOMAIN_NAME};
    return 301 https://$host$request_uri;
}
```

```
server {

    listen 443 ssl http2;

    server_name {DOMAIN_NAME};

    ssl_certificate /etc/letsencrypt/live/{DOMAIN_NAME}/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/{DOMAIN_NAME}/privkey.pem;

    include snippets/ssl-params.conf;
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
location / {
    proxy_pass http://{APP_NAME}/;
    proxy_http_version 1.1;
    proxy_cache_bypass $http_upgrade;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_max_temp_file_size 0;
    proxy_redirect off;
    proxy_read_timeout 240s;

    if ($http_origin ~* (^https?:/*.*\.example\.com$|^https?://localhost:\d+$)) {
        set $cors "true";
    }

    # See https://gist.github.com/algal/5480916
    # for CORS setup
    # OPTIONS indicates a CORS pre-flight request
    if ($request_method = 'OPTIONS') {
        set $cors "${cors}options";
    }

    # non-OPTIONS indicates a normal CORS request
    if ($request_method = 'GET') {
        set $cors "${cors}normal";
    }
    if ($request_method = 'PUT') {
        set $cors "${cors}normal";
    }
    if ($request_method = 'PATCH') {
        set $cors "${cors}normal";
    }
    if ($request_method = 'DELETE') {
        set $cors "${cors}normal";
    }
    if ($request_method = 'POST') {
        set $cors "${cors}normal";
    }

    # if it's a normal request, set the standard CORS responses header
    if ($cors = "truenormal") {
        add_header 'Access-Control-Allow-Origin' "$http_origin" always;
        add_header 'Access-Control-Allow-Credentials' 'true' always;
    }
}
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```

    add_header 'Access-Control-Expose-Headers' 'accesstoken, refreshtoken' always;
}

if ($cors = "trueoptions") {
    add_header 'Access-Control-Allow-Origin' "$http_origin";
    add_header 'Access-Control-Allow-Credentials' 'true';

    add_header 'Access-Control-Max-Age' 1728000;
    add_header 'Access-Control-Allow-Methods' 'GET, PUT, PATCH, DELETE, POST,
OPTIONS';
    add_header 'Access-Control-Allow-Headers' 'Authorization,Content-
Type,Accept,Origin,User-Agent,DNT,Cache-Control,X-Mx-ReqToken,Keep-Alive,X-Requested-With,If-
Modified-Since';

    add_header 'Content-Length' 0;
    add_header 'Content-Type' 'text/plain charset=UTF-8';
    return 204;
}
}
}
...

```

Change the following line:

```

...
if ($http_origin ~* (^https?:/*.*\.example\.com$|^https?://localhost:\d+$)) {
...

```

to use a regex for whatever domains you want to whitelist. In this example, we're allowing requests from any subdomain on `example.com` and requests from `localhost` on any port number, both on either `http` or `https`.

Also change this line:

```

...
'Access-Control-Expose-Headers' 'accesstoken, refreshtoken';
...

```

to list whatever response headers your app is setting which will be needed by the javascript apps calling your API.

Checkout this [annotated config file](<https://gist.github.com/algal/5480916>) to learn more about each of the additions we're making.

## [Step 11: BONUS! Control GPIO pins with Node.js](#)

Апаратне забезпечення та система управління для триколісного транспортного засобу  
One of the best things about using a Raspberry Pi is the ability to interact with the physical world using the [General Purpose Input/Output pins](<https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>). The RasPi 3 has 40 pins, 28 of which are programmable. You can wire these into buttons, switches, relays, servos, sensors, etc to do any sort of cool thing you can dream up. [pinout.xyz](<https://pinout.xyz/>) is a great resource to find out which pins do what.

There exist lots of Node.js libraries to help interact with these pins, but few are kept up to date and work with the latest RasPi models and newest versions of Node. However I found the [RPiIO](<https://www.npmjs.com/package/rpio>) package library and it is fantastic. It has an easy-to-understand interface, is super performant, and supports pretty much all RasPi models and most Node.js versions, including 6.x and 7.x.

To use it in your Node.js app run:

```
...  
yarn add rpio  
...
```

Before you can use the library in your Node.js app, you'll need to make sure the user running the Node script has the correct permissions.

Add the your user to the `gpio` group:

```
...  
sudo usermod -a -G gpio USERNAME  
...
```

And add a file to tell the interface to trust the `gpio` group:

```
...  
sudo nano /etc/udev/rules.d/20-gpiomem.rules  
...
```

and paste in:

```
...  
SUBSYSTEM=="bcm2835-gpiomem", KERNEL=="gpiomem", GROUP="gpio", MODE="0660"  
...
```

After restarting the Pi, you are ready to start controlling the pins with your app.

Include the library in your javascript file:

```
...
```

Апаратне забезпечення та система управління для триколісного транспортного засобу

```
const rpio = require('rpio');  
...  

```

Setting a pin as output and toggling it to high is as easy:

```
...  
rpio.open(12, rpio.OUTPUT, rpio.LOW);  
rpio.write(12, rpio.HIGH);  
...  

```

And reading the state of an input pin is just as easy:

```
...  
rpio.open(15, rpio.INPUT);  
const state = rpio.read(15);  
...  

```