

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Інститут енергетики, електроніки та електромеханіки

Кафедра промислової і біомедичної електроніки

Спеціальність 171 «Електроніка»

Освітня програма Електроніка

До захисту допускаю

Завідувач кафедри

С.Ю. Кривошеєв

(ініціали та прізвище)

(підпис, дата)

ДИПЛОМНА РОБОТА
другого (магістерського) рівня вищої освіти

Тема роботи: Мікроконтролерна система керування автономним пристроєм

Шифр роботи Е-М621 за 4.1

(група, номер теми за наказом)

Виконавець

Нікулін Максим Сергійович

(прізвище, ім'я, по-батькові)

Керівник

доц. Єресько Олександр В'ячеславович

(посада, прізвище, ім'я, по-батькові)

Харків 2022

[illegible]

ЗАТВЕРДЖУЮ

Завідувач кафедри

С.Ю. Кривошеєв

«___» _____ 20__ року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Нікуліну Максиму Сергійовичу

(прізвище, ім'я, по батькові)

1 Тема роботи Мікроконтролерна система керування автономним безпілотним пристроєм

керівник роботи Єресько Олександр В'ячеславович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «03» 10 2022 року № 1891 СТ

2 Строк подання студентом роботи 28.11.2022

3 Вихідні дані до роботи: Прилад використовує Li-ion акумулятор, два мотор-колеса та побудований на базі мікрокомп'ютера Raspberry Pi, модуль камери та Bluetooth, напруга живлення 12В, робоча температура -10...85 C

4 Перелік питань, які потрібно розробити у пояснювальній записці
Сформулювати технічне завдання, провести аналітичний огляд та розробити структурну схему, програмне забезпечення, алгоритми пошуку шляху, та розпізнавання перешкод. Виконати тестування програмного забезпечення. Провести дослід роботи системи. Розглянути питання економіки, охорони праці.

5 Перелік графічного матеріалу з точним зазначенням обов'язкових креслень
Презентація 10 аркушів А4

6 Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Техніко-економічне обґрунтування	Проскурня О.М., доцент.		
Охорона праці	Райко В.Ф., к.т.н.,професор.		

7 Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

Номер етапу	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Формулювання технічного завдання	03.10.2022	
2	Аналітичний огляд джерел, вибір методики досліджень	11.10.2022	
3	Розробка структурної та електричної схем	18.10.2022	
4	Розробка та тестування програмного забезпечення	01.11.2022	
5	Оформлення пояснювальної записки до ДР	21.11.2022	
6	Виконання презентації, написання доповіді	23.11.2022	
7	Подання ДР на відгук та зовнішню рецензію	24.11.2022	
8	Подання ДР на допуск до захисту	27.11.2022	
9	Захист ДР	01.12.2022	

Студент _____ Нікулін М.С.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Єресько О.В.
(підпис) (прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Інститут Енергетики, електроніки та електромеханіки
Кафедра Промислової і біомедичної електроніки
Спеціальність 171 «Електроніка»
Освітня програма «Електроніка»

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломної роботи

освітньо-кваліфікаційного рівня магістр

на тему: Мікроконтролерна система керування автономним пристроєм

Виконав студент 6 курсу, групи Е-М621 за
_____ Нікулін М.С.
(підпис, прізвище та ініціали)

Керівник _____ Єресько О.В.
(підпис, прізвище та ініціали)

Рецензент _____ Замаруєв В.В.
(підпис, прізвище та ініціали)

Нормоконтролер _____ Фетюхіна Л.В.
(підпис, прізвище та ініціали)

Харків 2022

РЕФЕРАТ

Пояснювальна записка до ДР 64с., 37 рис., 10 табл., 28 джерел

Ключові слова: ПОШУК ШЛЯХУ, АВТОПІЛОТ, НАВІГАЦІЯ, МІКРОКОНТРОЛЕР, АКСЕЛЕРОМЕТР, ГІРОСКОП

Об'єктом дослідження дипломного проекту магістра є рухомий пристрій з самостійною навігацією у просторі, на базі комбінованої навігаційної системи, розробка та розрахунок електричної принципової схеми та написання програмного забезпечення.

У процесі роботи були визначені існуючі способи керування комбінованими приладами, запропоновано структурну схему, виконано розробку електричної принципової схеми та розроблено програмну складову, яка забезпечує автоматичний пошук шляху в реальному часі з врахуванням перешкод.

Результати проектування рекомендуються для впровадження підприємствами електротехнічної промисловості різних форм власності.

Використання розробленої системи можливе на різних машинобудівних підприємствах для налагодження технологічних процесів, а також в навчальному процесі.

ABSTRACT

Explanatory note to DP 64p., 37 fig., 10 tables., 28 sources

Key words: PATH FINDING, AUTOPILOT, NAVIGATION, MICROCONTROLLER, ACCELEROMETER, GYROSCOPE

The object of research of the master's diploma project is a mobile device with independent navigation in space, based on a combined navigation system, the development and calculation of an electrical schematic diagram and the writing of software.

In the course of the work, the existing methods of controlling combined devices were determined, a structural diagram was proposed, an electrical schematic diagram was developed, and a software component was developed that provides automatic pathfinding in real time, taking into account obstacles.

Design results are recommended for implementation by enterprises of the electrical engineering industry of various forms of ownership.

The use of the developed system is possible at various machine-building enterprises for the adjustment of technological processes, as well as in the educational process.

ЗМІСТ

ВСТУП.....	3
1. ТЕХНІЧНЕ ЗАВДАННЯ.....	4
1.1 Найменування і область застосування	4
1.2 Підстава до розробки	4
1.3 Мета і призначення роботи.....	4
1.4 Технічні вимоги	4
1.5 Економічні показники	4
1.6 Стадії та етапи роботи	4
2. ВИБІР СПОСОБУ НАВІГАЦІЇ	5
2.1 Історія систем навігації.....	5
2.2 Обґрунтування способу навігації.....	7
2.2 Інерційна складова системи навігації	9
2.3 Активна складова системи навігації.....	12
3. ВИБІР ТА РОЗРАХУНОК СКЛАДОВИХ УЗЛІВ ПРИСТРОЮ	17
3.1 Вибір двигуна та драйвера	17
3.2 Розрахунок колес.....	25
3.3 Структурна схема пристрою	26
4 ВИБІР ТА ПРОГРАММУВАННЯ МІКРОКОНТРОЛЕРА.....	28
4.1 Вибір мікроконтролера.....	28
4.2 Програмування Raspberry Pi.....	29
5 ВИБІР АЛГОРИТМУ ПОШУКУ ШЛЯХУ	35
5.1 Огляд існуючих алгоритмів.....	36
5.2 Реалізація JPS алгоритму.....	38
6 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ	46
7 ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	56
ВИСНОВКИ.....	62
СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ	63

ВСТУП

Зростання кількості автомобілів на дорогах потребує впорядкування проблем безпеки, зменшення заторів на дорогах, що приводить до питань розумної мобільності. За класифікацією SAE International (Співтовариство автотранспортних інженерів), існує шість рівнів автоматизації водіння: від 0 (без автоматизації) до 5 (повна автоматизація автомобіля).

Зважаючи на постійно ускладнюючі сценарії тестування безпілотного транспорту і в міру наближення до комерційної експлуатації посилюються і вимоги до таких систем, що робить актуальною розробку нових і вдосконалення існуючих алгоритмів керування автономними транспортними засобами.

Навігація (лат. Navigatio, від лат. Navigo «пливу на судні») у широкому значенні – процес управління деяким об'єктом з метою переходу з точки у просторі А до точки Б. Стосовно мобільних наземних робіт під навігацією мають на увазі наступні завдання:

- визначення власного положення щодо деякої системи координат;
- вибір оптимального (за яким-небудь заданим критерієм) шляху проходження між двома положеннями в просторі;
- генерація управляючих впливів-команд для проходження оптимальним шляхом;
- запобігання зіткненням.

В багатьох країнах світу розробляються спеціальні закони та стратегії переходу на автономний транспорт, створюються стандарти до нормотворчої бази інтелектуальних засобів автономного водіння. Крім того, міжнародна юридична компанія dentons створила загальний індекс автономних транспортних засобів – інтерактивний онлайн-інструмент, який допомагає легко знайти основні норми безпілотного водіння по всьому світу й провести паралелі між різними юрисдикціями.

1 ТЕХНІЧНЕ ЗАВДАННЯ

1.1 Найменування і область застосування

Система навігації , призначена для орієнтування об'єкта в просторі.

1.2 Підстава до розробки

Розробка здійснюється в рамках навчального плану підготовки магістрів зі спеціальності «Електроніка» кафедри «Промислова і біомедична електроніка» у Національному технічному університеті «Харківський політехнічний інститут»

1.3 Мета і призначення роботи

Дипломна робота виконується з метою створення системи навігації безілотного пристрою

1.4 Технічні вимоги

В результаті проведення роботи необхідно обґрунтувати вибір структури електроприводу. Для моделі необхідно вибрати мікропроцесор для керування, а також розробити алгоритм пошуку шляху та розпізнавання перешкод.

1.5 Економічні показники

Основні техніко-економічні показники будуть знайдені на стадії техніко-економічного обґрунтування роботи.

1.6 Стадії та етапи роботи

Основні стадії і етапи роботи, терміни їх виконання приведені в бланку завдання на дипломну роботу.

2 ВИБІР СПОСОБУ НАВІГАЦІЇ

2.1 Історія систем навігації

Система навігації (двовимірна навігація) відома з давніх часів – у давні часи з використанням вимірювання положення сонця та зірок за допомогою різноманітних навігаційних приладів, які наведені на рисунку 2.1, а пізніше з розвитком науки та технологій навігація ускладнилася, а точність зросла. Вже час для проведення морських суден в Австралії в порт Карумба використовувався лазерний промінь опорного маяка, що задає напрямок прямолінійного руху, у «створ», в дуже вузькій кутовій просторовій зоні [1,2,3,4].



Рисунок 2.1 – Навігаційні прилади давнини

Безпілотне водіння кораблів та автомобілів привертають все більше уваги з боку наукової та інженерної спільноти та автомобільної індустрії. Перші прагнуть подолати технологічні бар'єри, що встають на шляху до повністю автономного наземного транспортного засобу, здатного приймати рішення у складних умовах міського руху. Другі працюють над стандартизацією отриманих рішень та їх частковим чи повним вбудовуванням у комерційні продукти. Останні працюють над законодавчими ініціативами, покликаними підготувати юридичну базу для майбутнього безпілотного транспорту [5].

Хоча історія розвитку навігації безпілотних автомобілів бере початок із 19-го

століття. Так наприклад у 1925 році компанія Houdina Radio Control продемонструвала радіокерований автомобіль, що пересувається по вулицях Нью-Йорка, а автомобіль Chandler, який зображено на рисунку 2.2. У 1926 році він був обладнаний приймально-передавальною антеною і керувався з машини, що йде за керованим авто. 1939 року на Нью-Йоркській Всесвітній виставці були представлені радіокеровані електромобілі, розроблені американським промисловим дизайнером-футуристом компанії Норманом Бел Гедесом, за підтримки компанії General Motors.

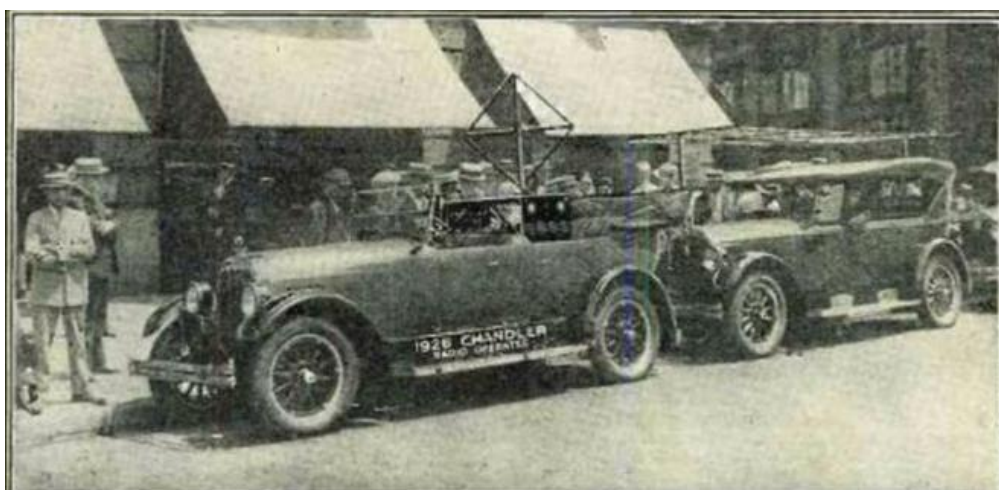


Рисунок 2.2 – Перший безпілотний автомобіль

Автомобілі рухалися електромагнітними хвилями, створюваними електричними пристроями, вбудованими в дорожнє полотно.

Зважаючи на постійно ускладнюючі сценарії тестування безпілотного транспорту і в міру наближення до комерційної експлуатації посилюються і вимоги до таких систем, що робить актуальною розробку нових і вдосконалення існуючих алгоритмів керування автономними транспортними засобами. Нарешті, у 2015 році компанія Tesla (Tesla Model S) представила систему допомоги водієві, що дозволяє здійснювати автономне керування автомобілем на шосе [6].

А вже у жовтні 2016 року, Tesla заявила, що всі їхні автомобілі збудовані на обладнанні, необхідному для забезпечення повної автономності водіння. Апаратне

забезпечення автомобілів Tesla включає вісім камер кругового огляду і дванадцять ультразвукових датчиків, на додаток до радарів, встановлених на автомобілі, як це зображено на рисунку 2.3.

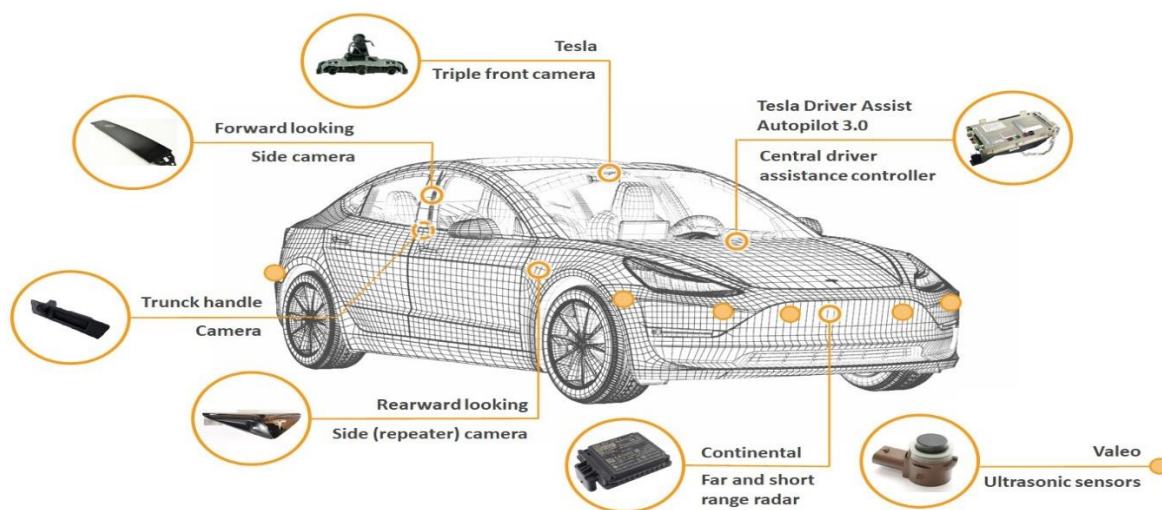


Рисунок 2.3 – Датчики у Tesla model 3

Програмна частина системи навігації об'єднує роботу як GPS координати точок маршруту прив'язані до докладних карт (Гугл – карт і т.п.), система навігації авто обладнана масою датчиків (енкодерами, акселерометрами, гіроскопами) які використовуються для «грубою» оцінки напрямку переміщення та пройденого шляху транспортним засобом.

Тим не менш, велика увага до проблеми з боку великих автомобільних компаній та дослідницьких лабораторій по всьому світу вселяє надію на досягнення амбітної мети, за винятком людського фактора з процесу пересування і, тим самим, збільшення його безпеки.

2.2 Обґрунтування способу навігації

Навігаційний комплекс – це сукупність приладів, алгоритмів і програмного забезпечення, що забезпечують орієнтування об'єкта в просторі (навігацію). До навігаційного комплексу можуть входити як складні навігаційні системи (наприклад, супутникова навігаційна система), так і окремі прилади, що дозволяють

визначити географічні координати об'єкта або його розташування відносно інших об'єктів. Виділяють два підходу до навігації: радіонавігація – заснована на взаємодії двох або більше спеціалізованих радіоприладів; інерціальна навігація – метод навігації (визначення координат і параметрів руху різних об'єктів – кораблів, літаків, ракет та ін) і управління їх рухом, що ґрунтується на властивостях інерції тіл, який є автономним, тобто не вимагає наявності зовнішніх орієнтирів або сигналів, що надходять ззовні.

Існують великі складності при навігації в замкнутому просторі – оточеному стінами наприклад тунель, підвал, завал тощо, коли немає зв'язку з GPS або аналогічною системою навігації, яка покладається на зовнішнє джерело сигналу. Для подібних випадків гарно зарекомендувала себе комбінована система навігації, яка поєднує інерційне числення шляху та пошук перешкод і коригування себе у просторі за допомогою різноманітних вимірювачів: доплерівських, візуальних та ультразвукових датчиків, QR кодів чи RFID метки, акселерометри та гіроскопів. Подібний підхід забезпечують високий рівень автономності безпілотної пристрою та дозволяє мінімізувати накопичення помилки.

Нижче, на рисунку 2.4 наведено загальний алгоритм руху пристрою:

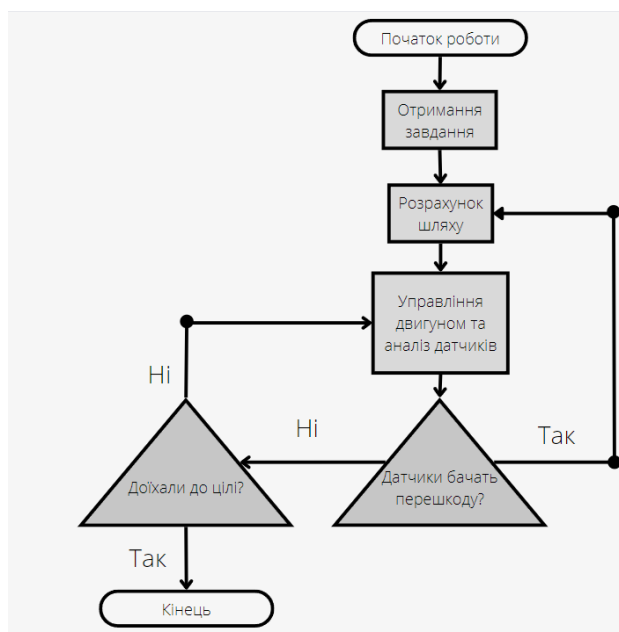


Рисунок 2.4 – Алгоритм руху безпілотної пристрою

Подібні системи використовуються на багатьох сучасних автомобілях з автопілотом, наприклад відома марка Tesla використовує широкий каскад датчиків положення у просторі.

2.2 Інерційна складова системи навігації

Інерційна навігація — метод навігації (визначення координат і параметрів руху різних об'єктів — кораблів, літаків, ракет та ін) і управління їх рухом, що ґрунтується на властивостях інерції тіл, який є автономним, тобто не вимагає наявності зовнішніх орієнтирів або сигналів, що надходять ззовні. Неавтономні методи вирішення завдань навігації ґрунтуються на використанні зовнішніх орієнтирів або сигналів (наприклад, зірок, маяків, радіосигналів тощо). Ці методи в принципі досить прості, але в ряді випадків не можуть бути здійснені через відсутність видимості або наявності перешкод для радіосигналів тощо. Необхідність створення автономних навігаційних систем явилася причиною виникнення інерціальних навігації, прикладом якої є зображений на рисунку 2.5 інерційний навігаційний пристрій французької балістичної ракети S3.

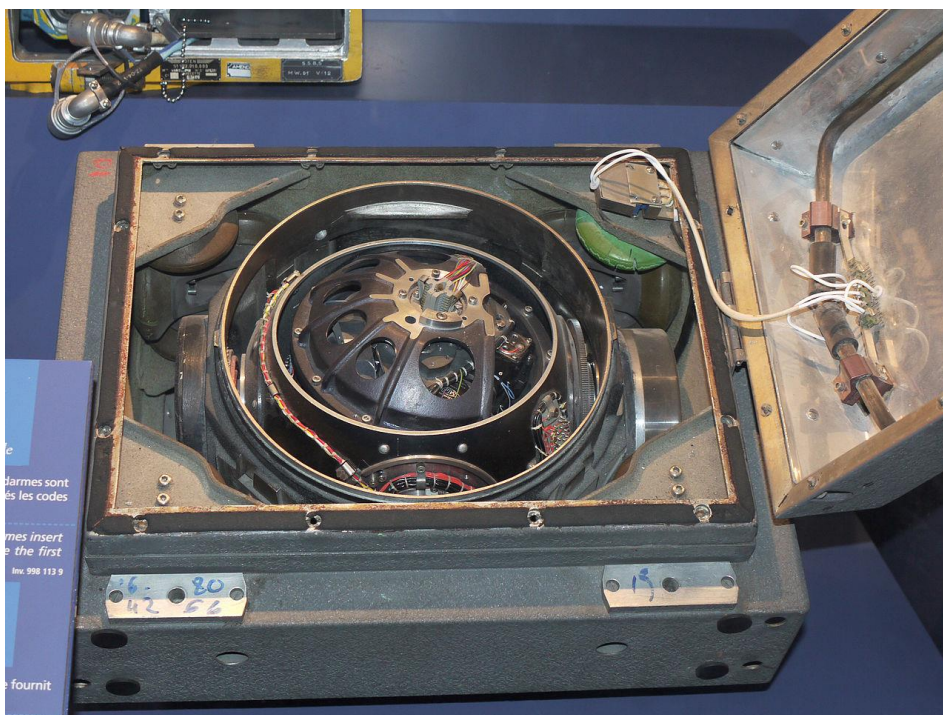


Рисунок 2.5 – Інерційний навігаційний пристрій французької балістичної ракети S3

Сутність інерційної навігації полягає у визначенні прискорення об'єкта та його кутових швидкостей за допомогою встановлених на об'єкті, що рухається, приладів і пристроїв, а за цими даними – розташування (координат) цього об'єкта, його курсу, швидкості, пройденого шляху та ін., а також у визначенні параметрів, необхідні стабілізації об'єкта та автоматичного управління його рухом. Це здійснюється за допомогою [2]:

- датчиків лінійного прискорення (акселерометрів);
- гіроскопічних пристроїв, що відтворюють на об'єкті систему відліку (наприклад, за допомогою гіростабілізованої платформи) і дозволяють визначати кути повороту та нахилу об'єкта, які використовуються для його стабілізації та керування рухом.
- обчислювальних пристроїв (ЕОМ), які за прискореннями (шляхом їх інтегрування) знаходять швидкість об'єкта, його координати та ін. Параметри руху.

В якості інерційної складової використаємо розповсюджене рішення: комбінований акселерометр та гіроскоп MPU6050 який зображено на рисунку 2.6:

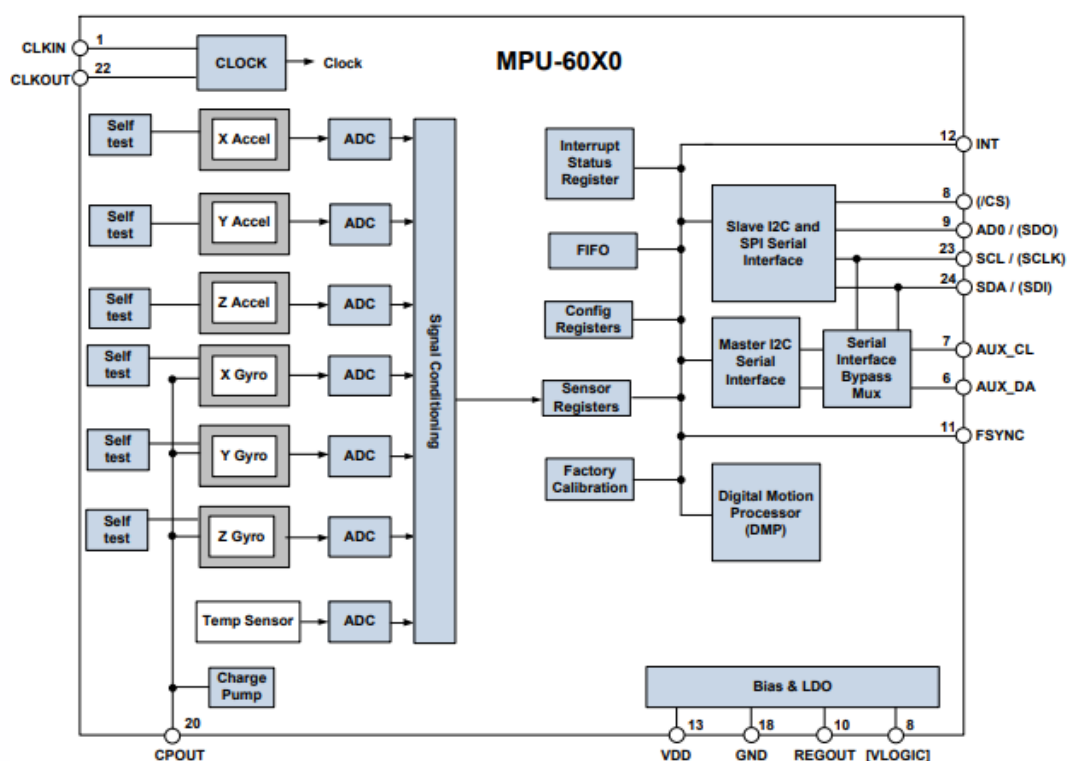


Рисунок 2.6 – Конструкція мікросхеми MPU6050

З рисунку 2.7. видно, що помилка фільтрації мало змінюється від частоти обурення, що діє модуль, і лише на великих частотах перевищує 0.5° . Зіставляючи результати експериментів з вимогами щодо точності вимірювань параметрів для систем управління, можна дійти висновку, що модуль MPU6050 забезпечує прийнятну точність на вирішення завдань управління пристроєм.

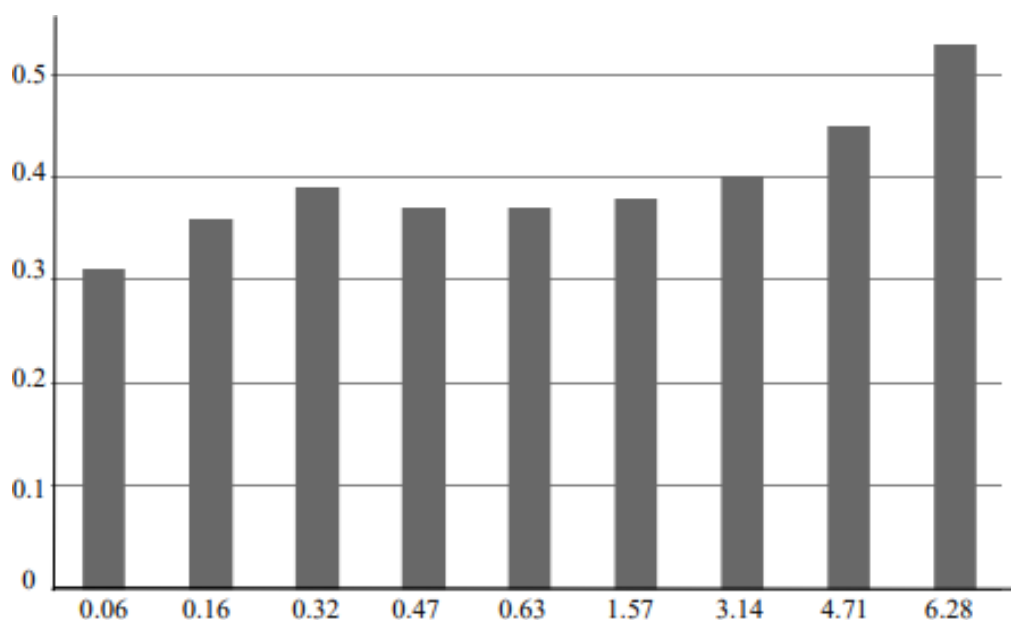


Рисунок 2.7 – Помилка фільтрації мікросхеми MPU6050.

Характеристики MPU6050:

- напруги живлення 2.375 – 3.46 вольт;
- споживаний струм до 4 мА;
- інтерфейс передачі даних – I2C;
- максимальна швидкість I2C – 400 кГц;
- вхід для інших датчиків I2C;
- внутрішній генератор на 8 МГц (поза модулем можливість підключити зовнішній кварцовий резонатор на 32.768 кГц або 19.2 МГц).

Функції MPU6050:

- трьох осьовий MEMS гіроскоп із 16 бітним АЦП;
- трьох осьовий MEMS акселерометр з 16 бітним АЦП;
- Digital Motion Processor (DMP);
- slave I2C для підключення до мікроконтролера;
- master I2C для підключення до мікросхеми додаткового датчика;
- регістри даних датчиків;
- FIFO;
- переривання;
- температурний сенсор;
- самоперевірка гіроскопа та акселерометра;
- регістр ідентифікації пристрою;

2.3 Активна складова системи навігації

Активні системи навігації використовують зовнішні джерела інформації для обчислення положення у просторі, такі як:

- радіонавігаційні системи, які дозволяють виконувати навігацію за допомогою аналізу радіовипромінювання зовнішніх джерел;
- супутникові системи, такі як GPS та ГЛОНАСС які дозволяють здійснювати навігацію відносно супутників;
- навігація за допомогою аналізу зображення з камер, зазвичай виконується за допомогою алгоритмів машинного навчання.

В якості активної складової навігаційної системи використаємо відеокамеру та програмний алгоритм розпізнавання об'єктів на базі нейронної мережі. Подібні алгоритми дуже гнучкі та легкі у підтримці.

Для початку, коротко розберемо, що являє собою завдання детектування об'єктів (object detection) на зображенні і які інструменти застосовуються для цього на сьогоднішній день. Приклад роботи інструменту для детектування об'єктів зображено на рисунку 2.8.

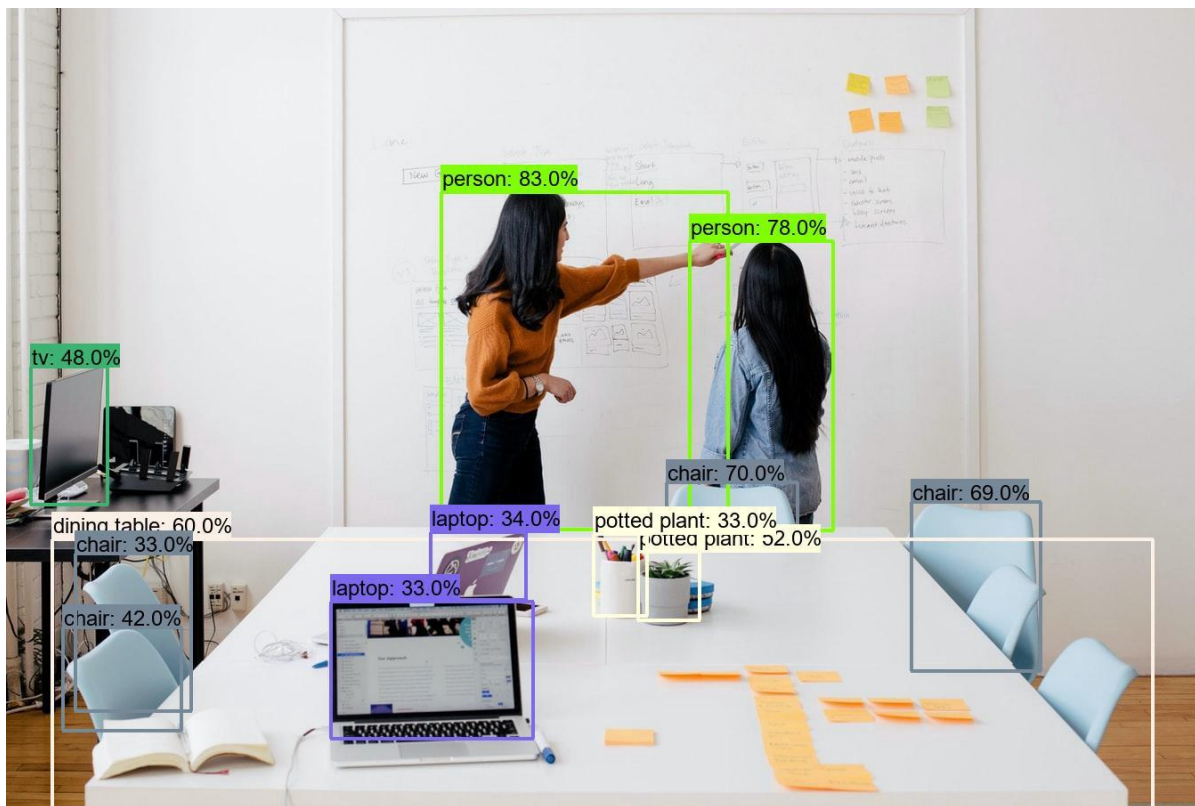


Рисунок 2.8 – Приклад детектування об'єктів

Зараз дуже багато завдань у галузі комп'ютерного зору вирішуються за допомогою згорткових нейронних мереж (Convolutional Neural Networks), надалі CNN. Завдяки своїй будові вони добре беруть ознаки із зображення. CNN використовуються у завданнях класифікації, розпізнавання, сегментації та ще у багатьох інших.

Нейронна мережа (або Штучна нейронна мережа) є математичною моделлю, а також її програмним або апаратним втіленням, побудованою за принципом організації та функціонування біологічних нейронних мереж — мереж нервових клітин живого організму, на рисунку 2.9 зображено вигляд . Задачією нейронної мережі є побудова функції по відомим точкам.

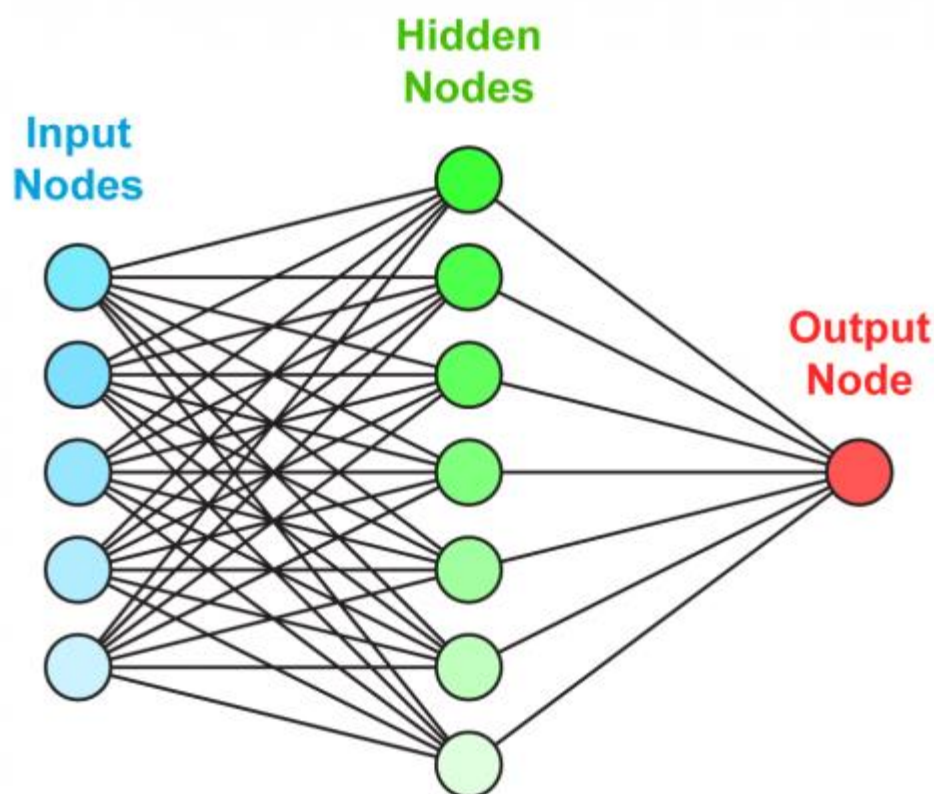


Рисунок 2.9 Структура нейронної мережі

Під архітектурою нейронної мережі розуміється її пристрій - послідовність нейронів та зв'язків між ними.

Популярні архітектури CNN для розпізнавання об'єктів:

1 R-CNN. Можна сказати перша модель для вирішення цього завдання. Працює як стандартний класифікатор зображень. На вхід мережі подаються різні регіони зображення і робиться передбачення. Дуже повільна, оскільки проганяє одне зображення кілька тисяч разів.

2 Fast R-CNN. Покращена і швидша версія R-CNN працює за схожим принципом, але спочатку все зображення подається на вхід CNN, потім з отриманого внутрішнього уявлення генеруються регіони. Але, як і раніше, досить повільна для завдань реального часу.

3 Faster R-CNN. Головна відмінність від попередніх у тому, що замість selective search алгоритму для вибору регіонів використовує нейронну мережу для

їхнього «заучування».

4 YOLO. Зовсім інший принцип роботи, порівняно з попередніми, не використовує регіони взагалі. На даний час це найшвидша мережа.

5 SSD. За принципом схожа на YOLO, але як мережа для отримання ознак використовує VGG16. Теж доволі швидка та придатна для роботи в реальному часі.

6 Feature Pyramid Networks (FPN). Ще один різновид мережі типу Single Shot Detector, через особливості вилучення ознак краще ніж SSD розпізнає дрібні об'єкти.

7 RetinaNet. Використовує комбінацію FPN+ResNet і завдяки спеціальній функції помилки (focal loss) дає більш високу точність (accuracy).

Так як безпілотний пристрій має не великі габарити, обмежений у обчислювальній потужності та повинен розпізнавати об'єкти у реальному часі доцільно використати архітектурне рішення yolov5 на базі фреймворка pytorch оскільки воно розроблено з ціллю використання на мобільних пристроях.

На рисунку 2.10 наведені характеристики різних архітектур нейронної мережі.

Model	size (pixels)	mAP ^{val} 0.5:0.95	mAP ^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	19.4	140.7	209.8
+ TTA	1536	55.8	72.7	-	-	-	-	-

Рисунок 2.10 – Параметри архітектур МЛ алгоритму yolov5

YOLO або You Only Look Once – це дуже популярна на даний момент архітектура CNN, яка використовується для розпізнавання багатьох об'єктів на зображенні. YOLOv5 – удосконалена п'ята версія YOLO, реалізована на фреймворку PyTorch. YOLOv5 вбудований в однойменний модуль для Python3, який можна встановити з рурі. Цей модуль надає дуже хорошу інфраструктуру як для навчання моделі, так і для тестування із побудовою графіків усіх ключових показників.

Робота нейронної мережі полягає у знаходженні об'єктів, які вона навчина знаходити у координатах зображення, як це показано на рисунку 2.11. Навчання нейронної мережі проходить за допомогою колекції підготовлених даних (зображень, на яких позначено цільові об'єкти).

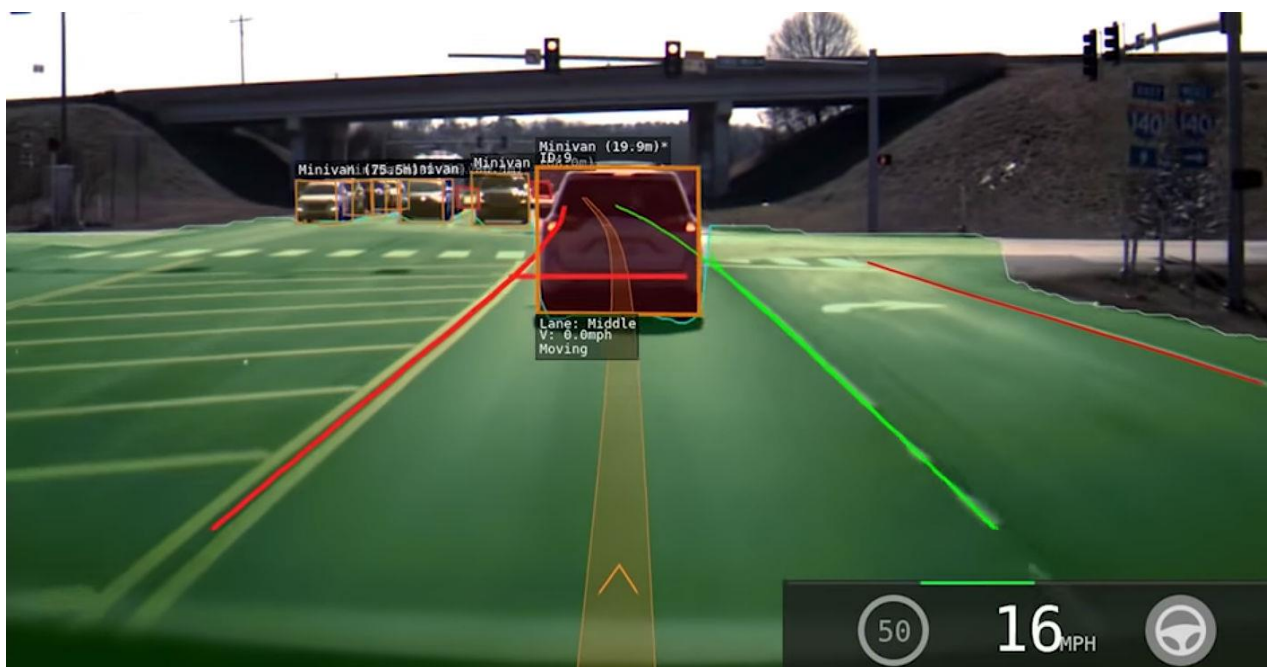


Рисунок 2.11 – Приклад використання МЛ алгоритму yolov5

3 ВИБІР ТА РОЗРАХУНОК СКЛАДОВИХ УЗЛІВ ПРИСТРОЮ

3.1 Вибір двигуна та драйвера

Скориставшись таблицею на рисунку 3.1, відповідно до технічного завдання та для забезпечення високої прохідності пристрою оберемо кроковий двигун.

Size 39mm Hybrid Stepping Motor Specifications

Model No.		Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque	# of Leads	Rotor Inertia	Weight	Detent Torque	Length
Single Shaft	Double Shaft	V	A	Ω	mH	g-cm		g-cm ²	kg	g-cm	mm
SY39ST20-0404A	SY39ST20-0404B	2.64	0.4	6.6	7.5	650	4	11	0.12	50	20
SY39ST20-0506A	SY39ST20-0506B	6.5	0.5	13	7.5	800	6	11	0.12	50	20
SY39ST34-0654A	SY39ST34-0654B	4.55	0.65	7.0	9.3	1800	4	20	0.18	120	34
SY39ST34-0404A	SY39ST34-0404B	12	0.4	30	32	2100	4				
SY39ST34-0604A	SY39ST34-0604B	9	0.6	15	16	2200	4				
SY39ST34-0306A	SY39ST34-0306B	12	0.3	40	20	1300	6				
SY39ST34-0166A	SY39ST34-0166B	12	0.16	75	50	1100	6				
SY39ST38-0504A	SY39ST38-0504B	12	0.5	24	45	2900	4	24	0.2	180	38
SY39ST38-0806A	SY39ST38-0806B	6	0.8	7.5	6	2000	6				

Рисунок 3.1 – Характеристик крокових двигунів від компанії
SongYong Hybrid Stepping Motor

В роботі використаємо кроковий двигун SY35ST26-0604A який має наступні характеристики:

- напруга живлення – 8 В;
- кут кроку – 1.8 гр;
- момент утримання – 0.51 Кг/см;
- габарити – 35.2x35.2x26 мм;
- опір обмотки – 5.4 Ом.

Оскільки він має високий обертаючий момент та може живитися від дванадцяти вольт, що дозволяє використовувати стандартну батарею без додаткових перетворювачів. Крім того його невеликі габарити, які приведені на рисунку 3.2 спрощують його монтаж на платформі безпілотної пристрою.

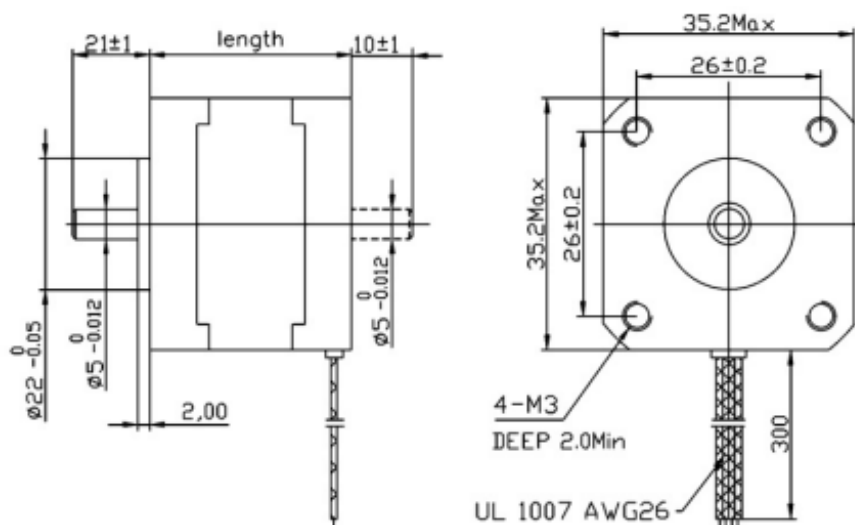


Рисунок 3.2 – Габарити крокового двигуна SY35ST26-0604A

Розрахуємо основні електричні параметри двигуна.

При напрузі $U_{W-StMot} = 8V$ та опорі обітки $R_{W-StMot} = 5,4\Omega$ струм в обмотці дорівнює:

$$I_{W-StMot} = U_{W-StMot} / R_{W-StMot} \quad (3.1)$$

$$U_{W-StMot} = 8 / 5,4 = 1.48 \text{ A}$$

Потужність що виділяється в обмотках крокового двигуна дорівнює:

Драйвер крокового двигуна дозволяє встановити максимальний струм обмотки двигуна на рівні 0,6A.

$$P_{W-StMot} = I_{W-StMot}^2 \cdot R_{W-StMot} \quad (3.2)$$

$$P_{W-StMot} = 0.6^2 \cdot 5,4 = 1,94W$$

При кроковому режимі керування тепло, що буде відповідати потужності що виділяється в двигуна, а поверхня охолодження двигуна дорівнює:

$$S_{\text{дв}} = 2 \cdot \text{Ш} \cdot \text{В} + 4 \cdot \text{Ш} \cdot \text{Д} \quad (3.3)$$

де: Ш – ширина = 35.2мм,

В – висота = 35.2мм,

Д – довжина = 26 мм

Таким чином:

$$S_{\text{дв}} = 2 \cdot 0.0352 \cdot 0.0352 + 4 \cdot 0.0352 \cdot 0.026 = 0.0069 \text{ м}^2$$

З одного квадратного метра поверхні можна зняти 625 W при перегріві 50°C. Звідси можна визначити перегрів двигуна:

$$T^{\circ}\text{C} = (P_{W-\text{StMot}} / 625) / S_{\text{дв}} \quad (3.3)$$

$$T^{\circ}\text{C} = (1,94 / 625) / 0.0069 = 0.51^{\circ}\text{C}$$

В якості драйвера даташит крокового двигуна SY35ST26-0604A [5] рекомендує використовувати драйвер A4988, схема використання та підключення якого зображені на рисунках 3.3 та 3.4 відповідно.

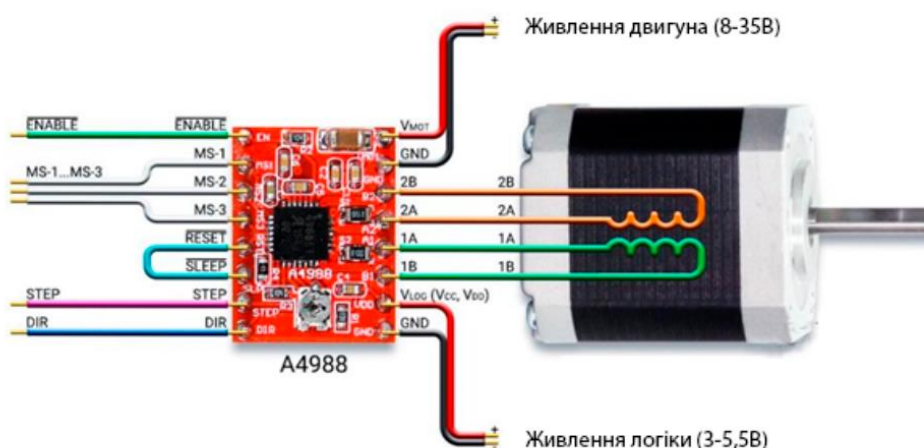


Рисунок 3.3 – Схема використання драйвера A4988

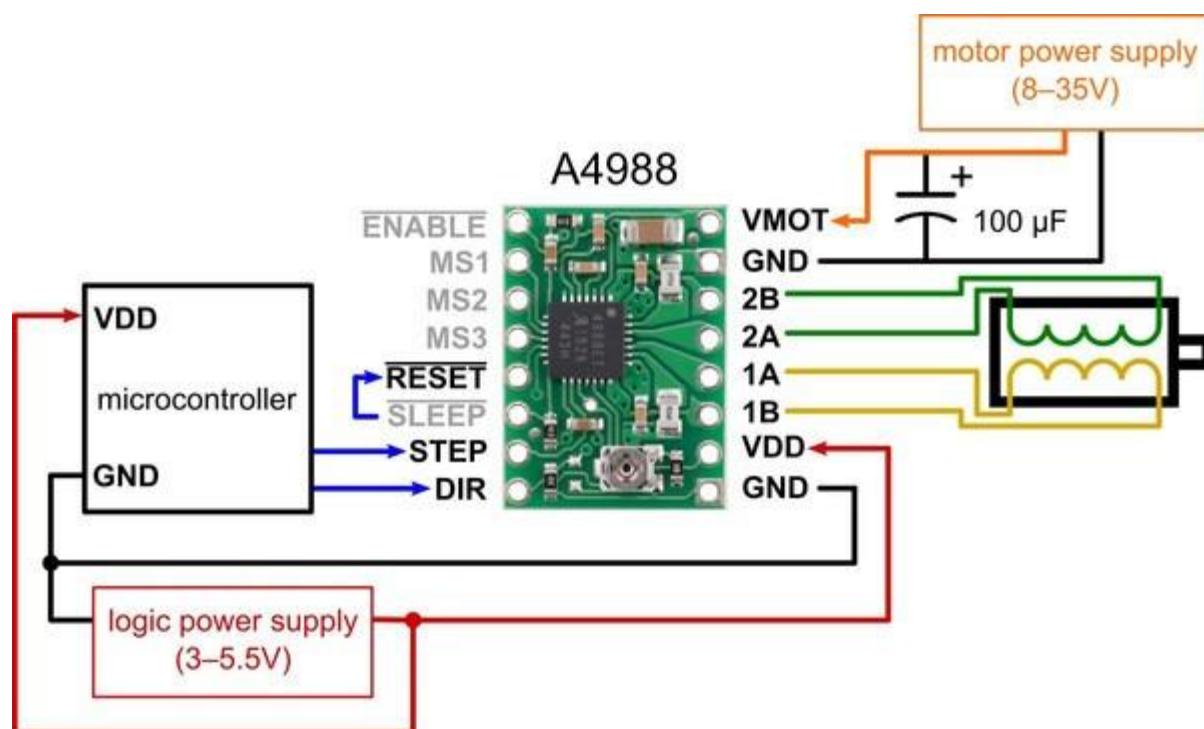


Рисунок 3.4 – Схема підключення драйвера A4988

- *ENABLE* – увімкнення/вимкнення драйвера;
- *MS1*, *MS2*, *MS3* – контакти для встановлення мікрокроку;
- *RESET* – скидання мікросхеми;

– *STEP* – генерація імпульсів для руху двигунів (кожен імпульс – крок), можна регулювати швидкість двигуна;

– *DIR* – встановлення напрямку обертання;

– *VMOT* – живлення двигуна (8 – 35 В);

– *GND* – загальний;

– *2B, 2A, 1A, 1B* – контакти для підключення обмоток двигуна;

– *VDD* – напруга живлення мікросхеми (3.5 – 5В).

Драйвер A4988 дозволяє використовувати режим кроку. Це досягається за рахунок подачі живлення на котушки з проміжними рівнями струму.

Наприклад, з кроковим двигуном NEMA 17 з кроком 1.8 градуса (200 кроків на оборот) в режимі 1/4 кроку, двигун видаватиме 800 мікрокроків на оборот. Режими роботи драйвера A4988 приведені на таблиці 3.1.

Таблиця 3.1 – Вибір режимів драйвера A4988

MS1	MS2	MS3	Розширення мікрокроку
0	0	0	Повний крок
1	0	0	1\2 кроку
0	1	0	1\4 кроку
1	1	0	1\8 кроку
1	1	1	1\16 кроку

Драйвер A4988 має три виведення селектора розміру кроку (режим), як це показано на рисунку 3.5, а саме: *MS1*, *MS2* та *MS3*. Встановивши відповідні логічні рівні на ці контакти, ми можемо налаштувати двигуни на один із п'яти режимів.

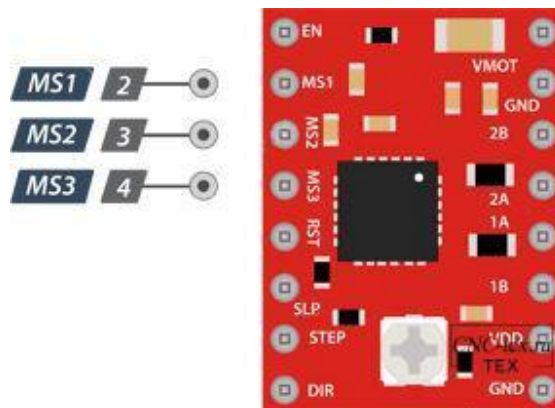


Рисунок 3.5 – Виведення вибору мікрокроху драйвер A4988

За замовчуванням ці три контакти підтягнуті до землі внутрішнім резистором. Якщо ми залишимо ці висновки не підключеними, двигун працюватиме в режимі повного кроку.

Крім того драйвер A4988 має два керуючі відводи, а саме: *STEP* і *DIR* які зображенні на рисунку 3.6.

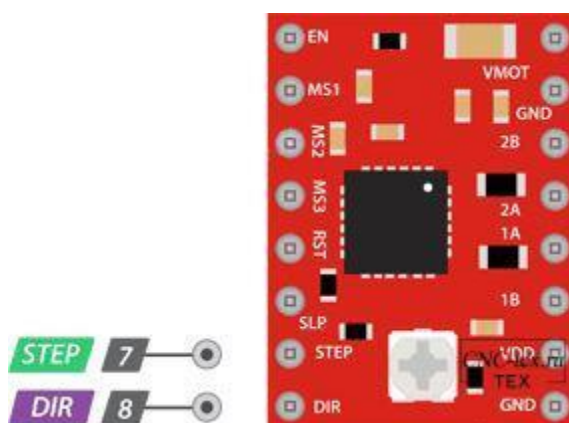


Рисунок 3.6 – Виведення керування драйвер A4988

STEP – керує мікрокроком мотора. Кожен високий імпульс, що відправляється на цей контакт, приводить двигун у дію на кількість мікрокроків,

задану висновками Microstep Selection (*MS1*, *MS2* та *MS3*). Чим швидше імпульси, тим швидше обертатиметься двигун.

DIR – керує напрямком обертання двигуна. Якщо на нього подати високий рівень, двигун буде обертатися за годинниковою стрілкою, а якщо низький – проти годинникової стрілки.

Якщо ви просто хочете, щоб двигун обертався тільки в одному напрямку, ви можете з'єднати висновок *DIR* безпосередньо з *VCC* або *GND* відповідно.

Виводи *STEP* та *DIR* не підтягнуті внутрішніми резисторами, тому ви не повинні залишати їх не підключеними. A4988 має три різні висновки для управління станом живлення, а саме. *EN*, *RST* та *SLP*, які зображені на рисунку 3.7.

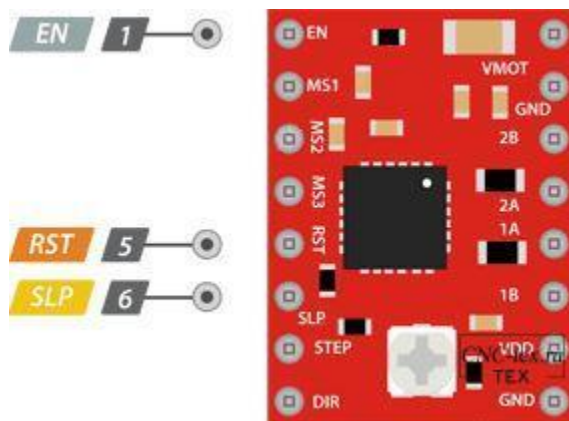


Рисунок 3.7 – Виведення керування драйвер A4988

EN – виведення (0)/вимкнення (1) драйвера A4988. За замовчуванням на цьому виводі встановлено низький рівень, тому драйвер завжди увімкнено.

SLP – подача на цей висновок сигналу низького рівня переводить драйвер в режим сну, зводячи до мінімуму споживання енергії. Ви можете використовувати це для економії енергії.

RST – при подачі сигналу низького рівня всі входні дані *STEP* ігноруються, доки не буде встановлено високий рівень. Низький рівень також скидає драйвер, встановлюючи внутрішній транслятор у визначений стан Home. Вихідний стан – це

в основному початкове положення, з якого запускається двигун, і воно відрізняється в залежності від дозволу мікрокроку.

Для підключення двигуна використовуються вихідні контакти: 1В, 1А, 2А та 2В. До цих контактів можна підключити будь-який біполярний кроковий двигун із напругою живлення від 8 до 35 В.

Кожен вихідний контакт модуля може забезпечити струм до 2 А. Однак величина струму, що подається на двигун, залежить від джерела живлення системи, системи охолодження та регулювання обмеження струму. Принципова схема драйвера з обв'язуванням зображена на рисунку 3.8.

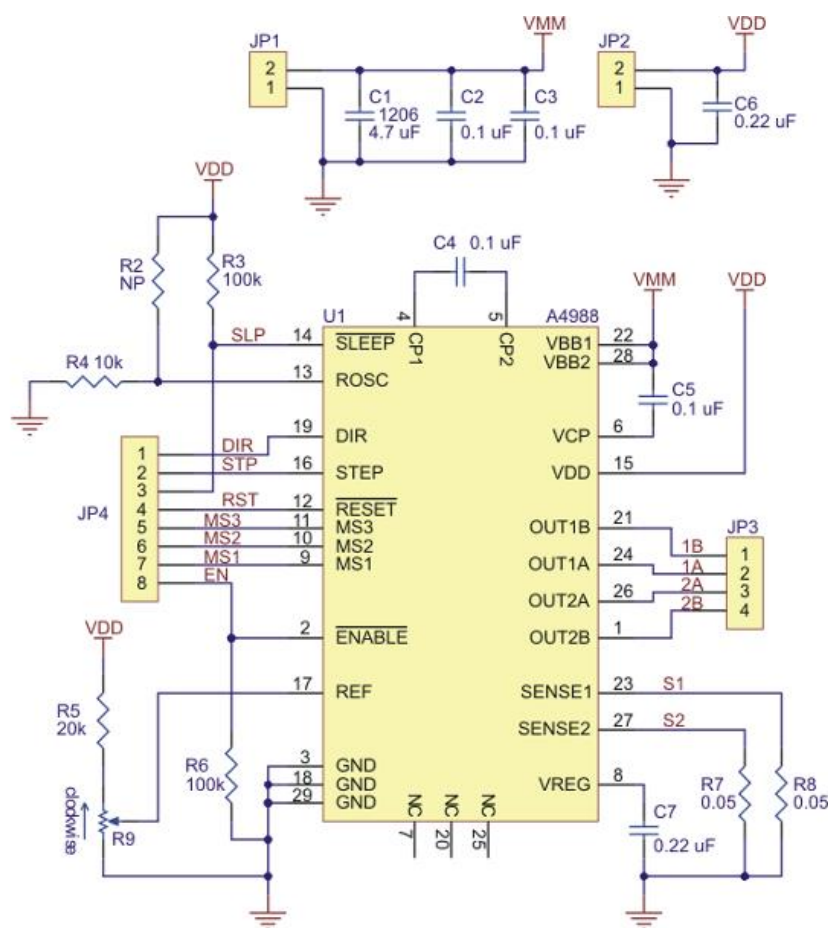


Рисунок 3.8 – Принципова схема використання драйвер A4988

3.2 Розрахунок колес

Для відслідковивання пройденного шляху з точки А до точки Б, окрім системи акселерометра та гіроскопа, використовується інформація про кількість обертів маршових колес. Це допомагає зменшити похибку інерційної системи. Для руху пристрою оберемо режим кроку 1/8 – це забезпечить високу точність керування та плавність руху пристрою при достатній швидкості.

Для поворота ротора маршового крокового двигуна на 360° (один оберт) для обраного режиму потрібно:

$$K_{1-об} = 360 / K_{шд} \quad (3.4)$$

де – $K_{шд} = 1.8/8$ (deg/step) кут одного кроку при напівкроковому режимі.

Підставимо значення да розрахуємо $K_{1-об}$:

$$K_{1-об} = 360 / (1.8/8) = 1600$$

Таким чином, при $R_{шд} = 32$ мм радіусі колеса маршового двигуна пройдений шлях за один оберт дорівнює:

$$L_{шд} = 2\pi \cdot R_{РП} \quad (3.5)$$

$$L_{шд} = 2\pi \cdot 32 = 201 \text{ мм}$$

Таким чином, відстань, яку пройде апарат за один крок двигуна буде дорівнювати:

$$L_{шд-1кр} = L_{шд} R_{РП} \quad (3.6)$$

$$L_{\text{шд-1kr}} = 201/1600 = 0.125 \text{ mm}$$

При найбільшій швидкості платформи $S_{\text{pl}} = 0,5$ м на сек (1,8 км на годину) двигун повинен виконати наступну кількість обертів:

$$K_{\text{об-дв}} = S_{\text{pl}} / L_{\text{шд}} \quad (3.7)$$

$$K_{\text{об-дв}} = 500/201 = 2.5 \text{ об/хв}$$

Кількість кроків, яка необхідна для одного оберта розраховується за формулою:

$$K_{\text{ш}_1\text{м/с}} = K_{1\text{-об}} \cdot K_{\text{об-дв}} \quad (3.8)$$

$$K_{\text{ш}_1\text{м/с}} = 1600 \cdot 2.5 = 4000 \text{ takt}$$

3.3 Структурна схема пристрою

На рисунку 3.6 зображена структурна схема пристрою, яка складається з наступних вузлів:

- джерело живлення;
- bluetooth модуль зв'язку;
- акселерометр та гіроскоп;
- мікроконтролер;
- відеокамера;
- дисплей;
- драйвера;
- крокові двигуни;

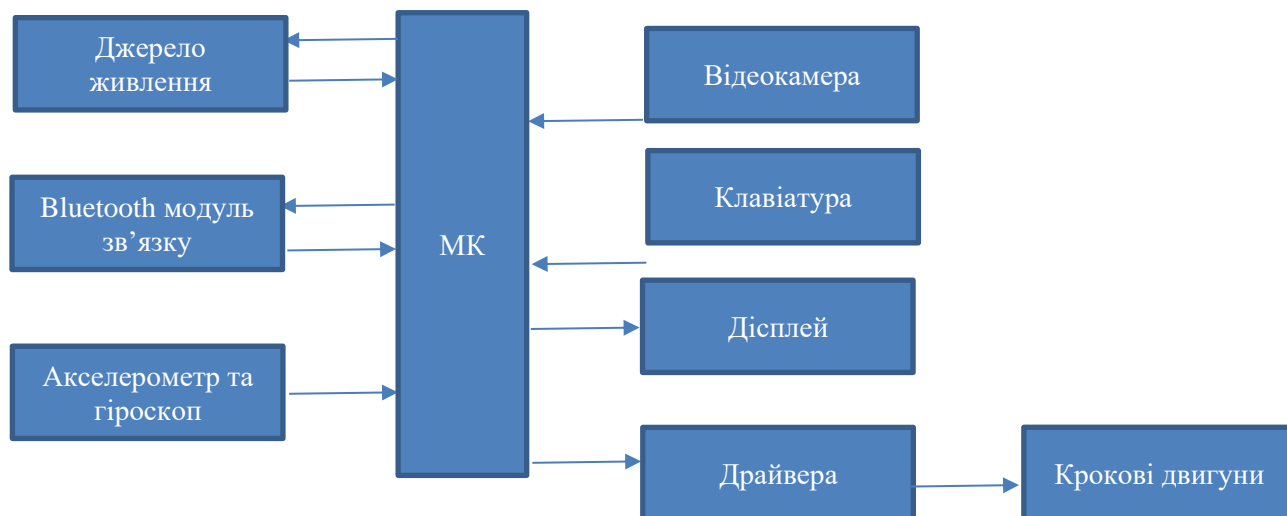


Рисунок 3.6 – Структурна схема керування автономним рухомим пристроєм

Для фізичного моделювання була обрана платформа до електроприводу якої входить два крокових двигуна один з них забезпечує переміщення платформи – мотор колесо, другий керує напрямком руху це рульовий вузол, який обертає мотор колесо вправо чи вліво режиму переміщення прямо – це положення фіксує контактний датчик положення заведення на вхід зовнішнього переривання мікроконтролера (МК). До електронного обладнання платформи входить мікроконтролер на стартер-кіт платі який організує виконання основного алгоритму, що забезпечує переміщення та пошук шляху, та монтажна плата на якій розміщено драйвера керування двигунами та роз'єм для підключення зовнішніх приладів, камери та датчиків. Допоміжними сервісними елементами пристрою є дісплей та клавіатура з чотирма кнопками (два курсори, Esc, Enter), яка за допомогою меню дозволяється керувати режимами роботи пристрою. До МК підключен вузол bluetooth зв'язку, який дозволяє, за необхідністю приєднати прилад до додатку на телефоні. Джерело живлення платформи має два Li-ion акумулятори, які з'єднані послідовно, забезпечує перетворення енергію акумуляторної батареї у напругу придатну для роботи мікроконтролера та інших приладів виконаних на мікросхемах.

4 ВИБІР ТА ПРОГРАММУВАННЯ МІКРОКОНТРОЛЕРА

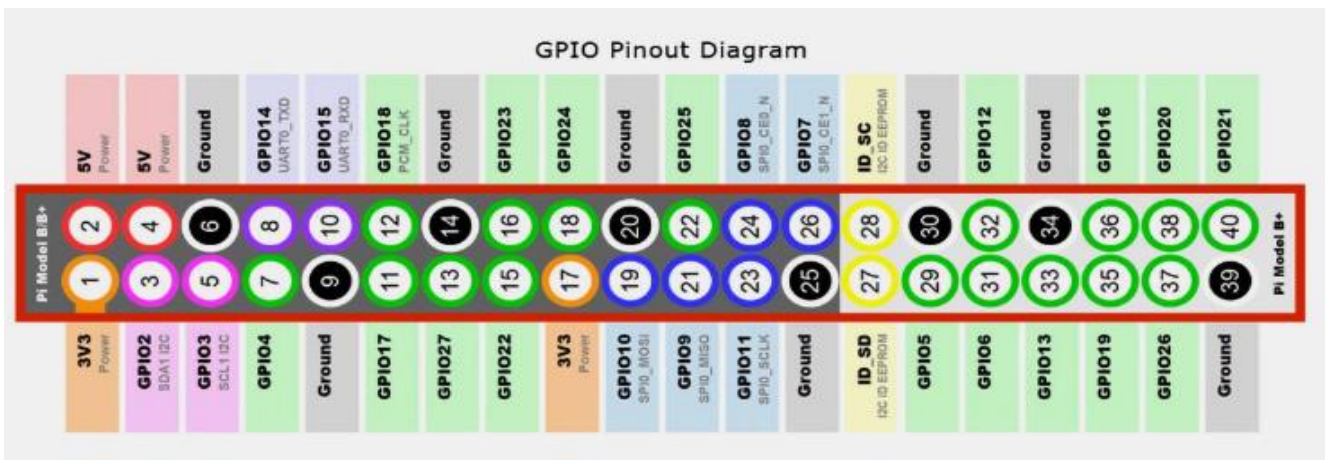
4.1 Вибір мікроконтролера

Для розгортання нейронної мережі необхідно мати мінімум 500мб оперативної пам'яті, крім того, для роботи алгоритму пошуку шляху в реальному часі та обробки даних з датчиків потрібен процесор на рівні сьогоденних мобільних пристроїв.

Цим критеріям підходить лінійка мініатюрних комп'ютерів Raspberry Pi. Окрім достатньої обчислювальної потужності, Raspberry Pi, в залежності від обраної моделі, має зручний інтерфейс для підключення відеокамери, зовнішніх елементів за допомогою відводів GPIO, та вбудований модуль Bluetooth, за допомогою якого можна під'єднатись до зовнішнього UI клієнта, наприклад, до мобільного телефону. Крім цього мікрокомп'ютери Raspberry підтримують використання високоабстрактних мов програмування, таких як Python або C#, що дозволяє спростити розробку складних програмних рішень.

Проаналізувавши характеристики різних моделей Raspberry Pi обираємо модель Raspberry Pi 3 A+ оскільки вона має достатній запас оперативної пам'яті (1GB), процесор Cortex-A53 розрядністю 64 біта та частотою 1.4 GHz. Для приєднання зовнішніх модулів платформа обладнена інтерфейсом введення/виведення загального призначення (англ. general-purpose input/output, GPIO) – це інтерфейс для зв'язку між компонентами комп'ютерної системи, наприклад мікропроцесором та різними периферійними пристроями. Контакти можуть бути як у ролі входу, і у ролі виходу – це, зазвичай, конфігурується. Контакти часто групуються у порти.

GPIO контакти не мають спеціального призначення та, як правило, залишаються невикористаними. Ідея у тому, що іноді системному інтегратору для побудови повної системи, використовує той чи інший чіп, може бути корисним мати кілька додаткових ліній цифрового управління. Розпіновка інтерфейсу GPIO для платформи Raspberry Pi A+ наведена на рисунку 4.1.



Риснок 4.1 – Схема контактів GPIO

Також платформа Raspberry Pi A+ має зручний інтерфейс для приєднання зовнішньої камери, наприклад широкоформатного модуля камери BauTech з наступними характеристиками:

- сумісність: з усіма версіями Raspberry Pi;
- сенсор: 5 MP OV5647;
- розмір камери: 1/4 дюйми;
- діафрагма: 2.35;
- фокусна відстань: регулюється;
- діагональ: 160 градусів;
- якість відео: 1080 p;
- розміри: 2,5 см x 2,4 см.

4.2 Программування Raspberry Pi

.NET nanoFramework – це безкоштовна платформа з відкритим вихідним кодом, заснована на .NET і призначена для малих пристроїв, що вбудовуються, мікроконтролерів. З її допомогою можна розробляти різні пристрої для Інтернету речей, пристрої, що носяться, наукові прилади, робототехнічні пристрої, можна створювати прототипи і навіть використовувати на промисловому обладнанні.

.NET nanoFramework є малою версією «великого» .NET Framework,

призначеного для настільних систем. Розробка програм ведеться мовою C# серед розробки Visual Studio. Сама платформа є виконавчим середовищем .NET коду, це дозволяє абстрагуватися від апаратного забезпечення та дає можливість переносити програмний код з одного мікроконтролера на інший, який також підтримує .NET nanoFramework. Програмний код C# для настільних систем, без змін або з невеликою адаптацією (необхідно пам'ятати про малий обсяг оперативної пам'яті) виконається на мікроконтролері. Завдяки цьому розробники на .NET з мінімальними знаннями в галузі мікроелектроніки зможуть розробляти різні пристрої на .NET nanoFramework.

Особливості .NET nanoFramework:

- 1) може працювати на 32– та 64-розрядних мікроконтролерах ARM, з наявністю всього 256 КБ флеш-пам'яті та 64 КБ ОЗУ;
- 2) працює на чіпі, в даний час підтримуються пристрої ARM Cortex-M і ESP32;
- 3) підтримує найпоширеніші інтерфейси такі як: GPIO, UART, SPI, I2C, USB, networking;
- 4) забезпечує вбудовану підтримку багатопоточності;
- 5) включає функції керування живленням для забезпечення енергоефективності, наприклад, пристроїв, що працюють від акумуляторних батарей;
- 6) підтримує суміщення керованого коду C# і некерованого коду C/C++ в одному проєкті;
- 7) автоматичне складання сміття завдяки збирачеві сміття.

Порівняно з іншими платформами:

Доступний інтерактивний налагоджувач при запуску коду на пристрої з точками зупинки.

Є розвинене та безкоштовне середовище програмування з Microsoft Visual Studio.

Підтримка великої кількості недорогих плат від різних виробників, включаючи: плати Discovery та Nucleo від ST Microelectronics, Quail від Mikrobus, Netduino від Wilderness Labs, ESP32 DevKit C, Texas Instruments CC3220 Launchpad, CC1352 Launchpad та NXP MX.

Легко переноситься на інші апаратні платформи та пристрої на ОС RTOS. В даний час сумісність забезпечується у відповідність до CMSIS і ESP32 FreeRTOS.

Цілком безкоштовне рішення з відкритим вихідним кодом, ніяких прихованих ліцензійних відрахувань. Від основних компонентів до утиліт, що використовуються для створення, розгортання, налагодження та компонентів IDE.

NuttX – це операційна система реального часу (RTOS), спеціально розроблена для використання в системах, що вбудовуються з мікроконтролерами або процесорами розрядністю від 8 до 32 біт. Завдяки гнучкій можливості налаштування NuttX, розробник може включати в образ лише ті модулі, які дійсно потрібні для проекту. Одним із головних принципів NuttX є відповідність стандартам POSIX та ANSI. Завдяки цьому визначається загальний інтерфейс для різних операційних систем, що сприяє переносимості, повторному використанню коду та підтримки додатків, що використовують POSIX та ANSI. Додаткові стандартні API з Unix та інших RTOS (така як VxWorks) адаптовані для функціональності, недоступні у відповідності до стандартів, та низькорівневим оточенням для вбудованих систем (як fork()).

Список платформ, що підтримуються, досить великий, до нього входять процесори ARM Cortex, Atmel AVR, Intel, RISC-V, STM32, і т.д. Серед підтримуваних МК є серія ESP32 від Espressif. Сама компанія Espressif активно інвестує у проект NuttX.

NuttX розповсюджується під вільною ліцензією Apache 2.0 License.

Образ NuttX для RP2040 підтримує інтерфейси UART, I2C, SPI, PIO (RP2040 Programmable I/O) тощо. Залежно від модулів, що підключаються, будуть доступні LCD екрани: ssd1306 (I2C), lcd1602 (I2C), st7735 (SPI). Підтримується мережа Ethernet (модуль ENC28J60 на SPI).

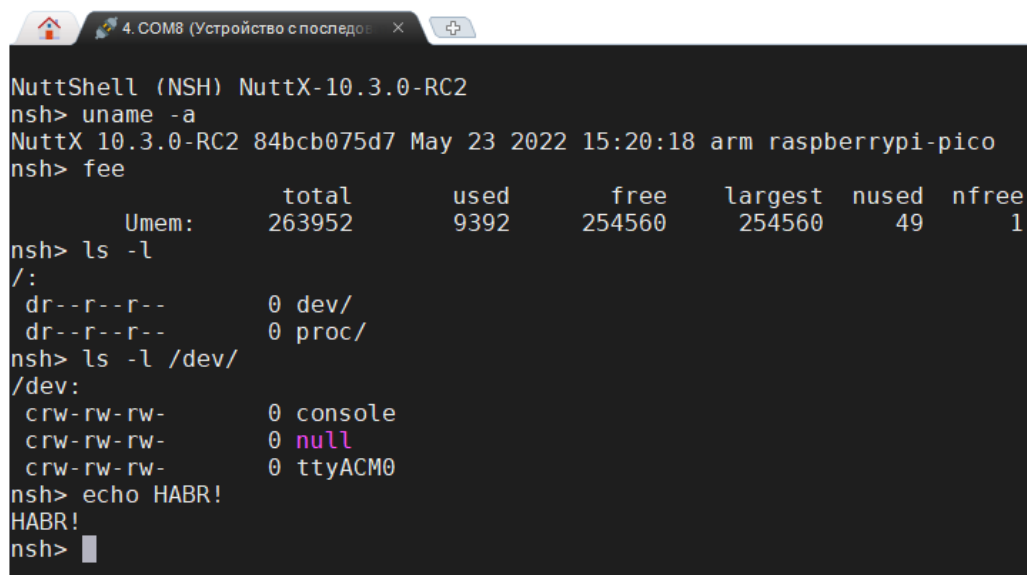
Зберемо мінімальний образ для RP2040 з NuttShell в Ubuntu 20.04, параметр для складання – usbnsn. NuttShell це аналог консолі в Linux. Параметр usbnsn використовує порт USB CDC/ACM для роботи з NuttShell. Скрипт зборки зображено на рисунку 4.2.

```
$ sudo apt-get update
$ sudo apt-get install -y git
$ sudo apt-get install -y \
bison flex gettext texinfo libncurses5-dev libncursesw5-dev \
gperf automake libtool pkg-config build-essential gperf genromfs \
libgmp-dev libmpc-dev libmpfr-dev libisl-dev binutils-dev libelf-dev \
libexpat-dev gcc-multilib g++-multilib picocom u-boot-tools util-linux
$ sudo apt-get install -y kconfig-frontends
$ sudo apt-get install -y gcc-arm-none-eabi binutils-arm-none-eabi
$ sudo reboot now
$ mkdir -p /usr/share/pico
$ cd /usr/share/pico
$ git clone https://github.com/raspberrypi/pico-sdk
$ cd /usr/share/pico/pico-sdk
$ git reset --hard 26653ea81e340
$ export PICO_SDK_PATH=/usr/share/pico/pico-sdk
$ mkdir -p /usr/share/pico/nuttxspace
$ cd /usr/share/pico/nuttxspace
$ git clone https://github.com/apache/incubator-nuttx.git nuttx
$ git clone https://github.com/apache/incubator-nuttx-apps apps
$ cd /usr/share/pico/nuttxspace/nuttx
$ ./tools/configure.sh -l raspberrypi-pico:usbnsn
$ make -j
```

Рисунок 4.2 – Скрипт зборки NuttX

Після компіляції /usr/share/pico/nuttxspace/nuttx з'явиться файл nuttx.uf2.

Для завантаження прошивки в Raspberry Pi A+ необхідно на платі натиснути кнопку BOOTSEL і підключити плату по USB порту. Після цього можливо відкрити NuttShell консоль, яка зображена на рисунку 4.3 та завантажити тестовий скрипт мерехтіння вбудованим світлодіодом, який зображено на рисунку 4.4.



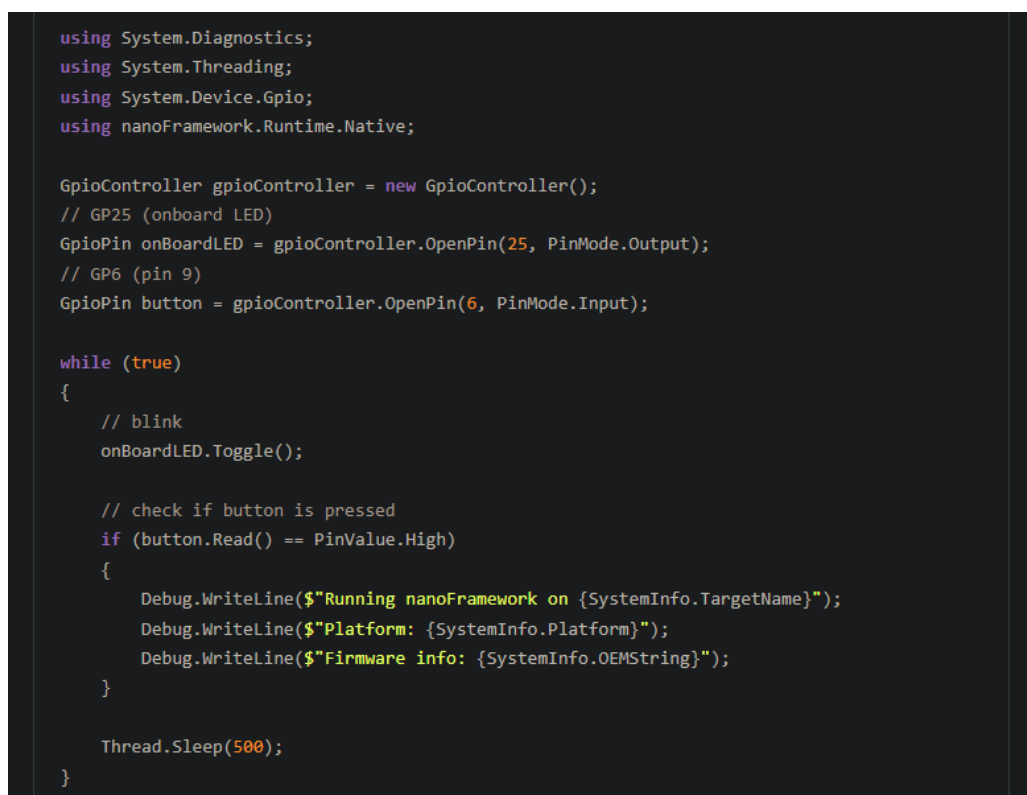
```

NuttShell (NSH) NuttX-10.3.0-RC2
nsh> uname -a
NuttX 10.3.0-RC2 84bcb075d7 May 23 2022 15:20:18 arm raspberrypi-pico
nsh> free
              total        used        free     largest     nused     nfree
Umem:         263952         9392       254560       254560         49         1
nsh> ls -l
/:
dr--r--r--    0 dev/
dr--r--r--    0 proc/
nsh> ls -l /dev/
/dev:
crw-rw-rw-    0 console
crw-rw-rw-    0 null
crw-rw-rw-    0 ttyACM0
nsh> echo HABR!
HABR!
nsh>

```

Рисунок 4.3 – NuttShell консоль ОС NuttX на Raspberry Pi

.NET nanoFramework може працювати з неймовірними мінімальними вимогами, для запуску необхідно лише 256 КБ флеш-пам'яті та 64 КБ ОЗУ. Raspberry Pi A+ цілком задовольняє цим вимогам.



```

using System.Diagnostics;
using System.Threading;
using System.Device.Gpio;
using nanoFramework.Runtime.Native;

GpioController gpioController = new GpioController();
// GP25 (onboard LED)
GpioPin onBoardLED = gpioController.OpenPin(25, PinMode.Output);
// GP6 (pin 9)
GpioPin button = gpioController.OpenPin(6, PinMode.Input);

while (true)
{
    // blink
    onBoardLED.Toggle();

    // check if button is pressed
    if (button.Read() == PinValue.High)
    {
        Debug.WriteLine($"Running nanoFramework on {SystemInfo.TargetName}");
        Debug.WriteLine($"Platform: {SystemInfo.Platform}");
        Debug.WriteLine($"Firmware info: {SystemInfo.OEMString}");
    }

    Thread.Sleep(500);
}

```

Рисунок 4.4 – Тестовий скрипт мерехтіння вбудованим світлодіодом на платі Raspberry Pi

Таким чином можливо абстрагуватись від розробки під конкретну систему, зробити код легким для читання та розуміння людиною.

Програмне забезпечення для керування безпілотним пристроєм складається з наступних модулів:

- Модуль імпліmentaції нейронної мережі та допоміжні класи – який виконує функцію розпізнавання елементів на зображенні;
- Модуль навігації та допоміжні класи – який виконує функцію збору даних з зовнішніх датчиків (акселерометр, гіроскоп, інформація про кількість обертів колес);
- Модуль пошуку шляху з урахування перешкод;
- Модуль зв'язку – виконує функцію передачі інформації користувачеві через Bluetooth;
- Модуль керування двигунами пристрою.

Структура проекту зображена на рисунку 4.5

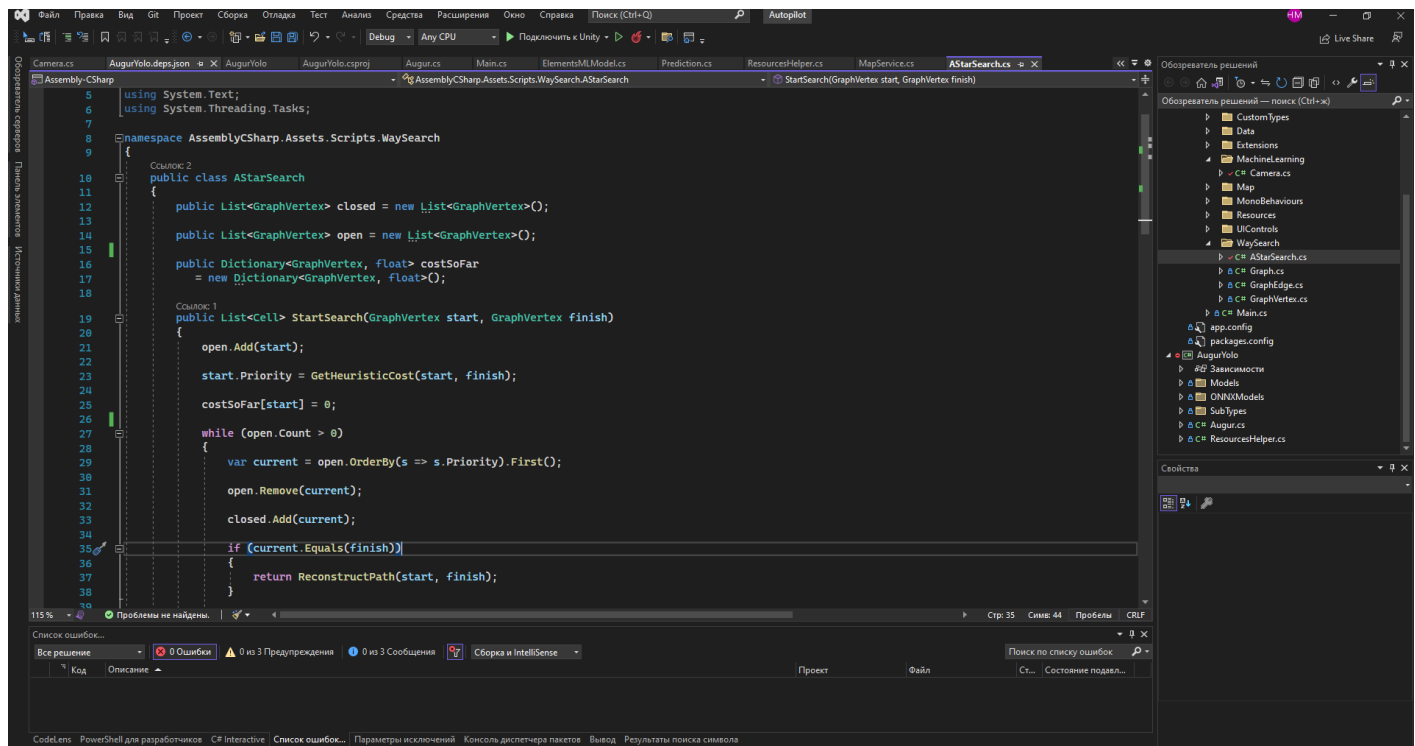


Рисунок 4.5 – Програмне забезпечення розроблене в IDE visual studio

5 ВИБІР АЛГОРИТМУ ПОШУКУ ШЛЯХУ

Високі темпи інформатизації різних видів діяльності в даний час привели до того, що з'явилася можливість комп'ютерного моделювання та проектування складних систем, вивчення їх властивостей і управління ними в умовах дефіциту часу, обмеженості ресурсів, неповноти інформації. Однак для дослідження характеристик будь-якої системи математичними методами повинна бути обов'язково виконана формалізація, тобто побудована математична модель. Дослідження за допомогою математичних моделей найчастіше є єдино можливим способом вивчення складних систем і вирішення найважливіших практичних завдань управління. Графи виявилися гарною математичною моделлю широкого класу об'єктів і процесів. Теорія графів застосовується в таких областях, як фізика, хімія, теорія зв'язку, проектування ЕОМ, електроніка, машинобудування, архітектура, дослідження операцій, генетика, психологія, соціологія, економіка, антропологія і лінгвістика. При цьому зазвичай на графі вирішуються завдання пошуку оптимального маршруту, досяжності, завдання мережевого планування, потокова задача.

Зараз теорія графів охоплює велику кількість матеріалу та активно розвивається в багатьох напрямках. Нас же цікавить задача пошуку маршруту. Тому при описі алгоритмів та методів зупинимося на тих, що стосуються маршрутизації. Побудова математичного визначення графу здійснюється шляхом формалізації «об'єктів» та «зв'язків» як елементів деяких множин.

Задача про найкоротший шлях полягає у знаходженні найкоротшого шляху від заданої початкової вершини до заданої кінцевої вершини. Формулювання задач про знаходження відстаней таке:

- для заданої початкової вершини a знайти найкоротші шляхи від a до всіх інших вершин;
- знайти найкоротші шляхи між усіма парами вершин.

Виявляється, що майже всі методи розв'язання задачі про найкоротший шлях

від заданої початкової вершини до заданої кінцевої вершини, також дають змогу знайти й найкоротші шляхи від вершини A до всіх інших вершин графа. Отже, за їх допомогою можна розв'язати задачу 1 із невеликими додатковими обчислювальними витратами. З іншого боку, задачу 2 можна розв'язати або n разів застосувавши алгоритм задачі 1 із різними початковими вершинами, або один раз застосувавши спеціальний алгоритм.

Розглянемо найросповсюдженіші алгоритми пошуку шляху.

5.1 Огляд існуючих алгоритмів

5.1.1 Пошук завширшки (BFS, Breadth-First Search)

Алгоритм був розроблений незалежно Муром і Лі для різних додатків (пошук шляху в лабіринті та розведення провідників відповідно) у 1959 та 1961 роках. Цей алгоритм можна порівняти з запалюванням сусідніх вершин графа: спочатку ми запалюємо одну вершину (ту, з якої починаємо шлях), а потім вогонь за один елементарний проміжок часу перекидається на всі сусідні вершини, що з нею не горять. Згодом те саме відбувається з усіма підпаленими вершинами. Таким чином, вогонь поширюється «завширшки». В результаті його роботи буде знайдено найкоротший шлях до потрібної клітини.

5.1.1 Алгоритм Дейкстри (Dijkstra)

Цей алгоритм названо на ім'я творця і було розроблено 1959 року. У процесі виконання алгоритм перевірить кожну з вершин графа і знайде найкоротший шлях до вихідної вершини. Стандартна реалізація працює на зваженому графі – графі, у якого кожен шлях має вагу, тобто. "вартість", яку треба буде "заплатити", щоб перейти по цьому ребру. При цьому у стандартній реалізації ваги невід'ємні. На картатому полі вага кожного ребра графа приймається однаковим (наприклад, одиницею).

5.1.1 A* (A «із зірочкою»)

Вперше описаний у 1968 році Пітером Хартом, Нільсом Нільсоном та Бертрамом Рафаелем. Даний алгоритм є розширенням алгоритму Дейкстри,

прискорення роботи досягається за рахунок евристики – при розгляді кожної окремої вершини перехід робиться в ту сусідню вершину, імовірний шлях з якої до найвищої вершини найкоротший. У цьому є безліч різних методів підрахунку довжини передбачуваного шляху з вершини. Результатом роботи також буде найкоротший шлях.

5.1.1 Пошук за першим найкращим збігом (Best-First Search)

Удосконалена версія алгоритму пошуку завширшки, що відрізняється від оригіналу тим, що в першу чергу розгортаються вузли, шлях з яких до кінцевої вершини імовірно коротший. Тобто. за рахунок евристики робить для BFS те, що A* робить для алгоритму Дейкстри.

5.1.1 IDA* (A* з ітеративним поглибленням)

Розшифровується як Iterative Deeping A*. Є зміненою версією A*, яка використовує менше пам'яті за рахунок меншої кількості вузлів, що розгортаються. Працює швидше за A* у разі вдалого вибору евристики. Результат роботи – найкоротший шлях.

5.6.1 Jump Point Search

Наймолодший із перерахованих алгоритмів був представлений у 2011 році. Являє собою вдосконалений A*. JPS прискорює пошук шляху, «перестрибуючи» багато місць, які потрібно переглянути. На відміну від подібних алгоритмів JPS не вимагає попередньої обробки та додаткових витрат пам'яті.

Оскільки JPS алгоритм на даний час є вершиною еволюції алгоритмів пошуку шляху – оберемо його для нашої роботи.

5.2 Реалізація JPS алгоритму

Цей алгоритм є покращеним алгоритмом пошуку шляху A*. JPS прискорює пошук шляху, “перестрибуючи” багато місць, які потрібно переглянути. На відміну від подібних алгоритмів JPS не вимагає попередньої обробки та додаткових витрат пам'яті. Даний алгоритм представлений у 2011 році, а у 2012 отримав високі відгуки. Приклад використання алгоритму зображено на рисунку 5.1.

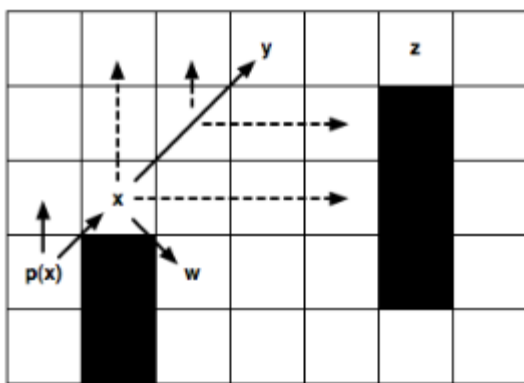


Рисунок 5.1 – Задача, в якій використовується алгоритм

Алгоритм працює на неорієнтованому графі єдиної вартості. Кожне поле карти має ≤ 8 сусідів, які можуть бути прохідні чи ні. Кожен крок у напрямку (вертикалі або по горизонталі) має вартість 1; крок по діагоналі має вартість $\sqrt{2}$. Рухи через перешкоди заборонені. Позначення відноситься до одного з восьми напрямків руху (вгору, вниз, вліво і т.д.).

– Запис $y = x + kd$ означає, що точка y може бути досягнута через k кроків з x у напрямку d . Коли d – рух по діагоналі, переміщення поділяється на два переміщення прямою $d1$ і $d2$.

– Шлях $p = (n0, n1, \dots, nk)$ – упорядковане переміщення по точках без циклів із точки $n0$ до точки nk .

– Позначення $p \setminus x$ означає, що точка x не зустрічається на шляху p .

– Позначення $len(p)$ означає довжину чи вартість шляху p .

– Позначення $dist(x, y)$ означає довжину чи вартість шляху між точками x та y .

5.2.1 Jump points

"Стрибкові точки" дозволяють прискорити алгоритм пошуку шляху, розглядаючи тільки "необхідні" точки. Такі точки можуть бути описані двома простими правилами вибору сусідів під час рекурсивного пошуку: одне правило для

прямолінійного руху та інше – для діагонального. В обох випадках необхідно довести, що виключаючи з набору найближчих сусідів навколо точки, знайдеться оптимальний шлях із предка поточної точки до кожного із сусідів, і цей шлях не міститиме відвідану точку. Розглянемо випадок 1 зображений на рисунку 5.2, який відбиває основну ідею:

Випадок 1: Відсічений сусід:

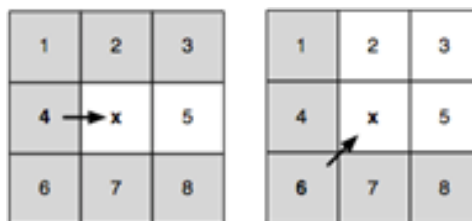


Рисунок 5.2 – Випадки застосування відсічення сусідів

X – поточна розглянута точка. Стрілка вказує напрямок руху. І там і там можна відразу відсікти сусідів, виділених сірим, т.к. туди можна потрапити оптимальним шляхом з $p(x)$, ніколи не проходячи через x .

Будемо посилалися на безліч точок, що залишаються після відсікання справжніх сусідів поточної точки. Вони позначені білими малюнку. В ідеалі ми хочемо враховувати тільки справжніх сусідів під час перегляду. Тим не менш, у деяких випадках, наявність перешкод може означати, що ми повинні також розглянути невеликий набір до K додаткових точок ($0 \leq K \leq 2$). Ми говоримо, що це точки вимушених сусідів поточної позиції, як показано на рисунку 5.3.

Випадок 2: Вимушений сусід:

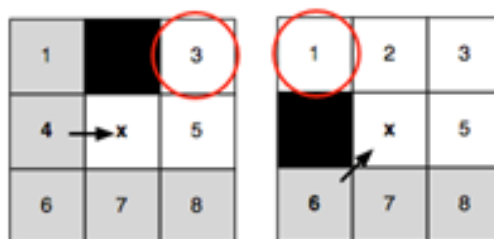


Рисунок 5.3 – Приклад вимушеного сусіда

наступник стрибкової точки x . Ця точка цікава тим, що є сусід z , в який можна потрапити оптимальним шляхом тільки через y . Проміжні точки не генеруються та не розглядаються.

Права частина: Рекурсивно приймаємо діагональні правила відсічення. Зверніть увагу, що перед кожним наступним діагональним кроком необхідно рекурсивно пройтися прямими лініями (виділені пунктиром). Тільки якщо обидві "прямі" рекурсії не можуть визначити точку наступного стрибка, то рухаємося далі по діагоналі. Точка w – вимушений сусід x , створюється як звичайний.

Далі опишемо, яким чином відсікати безліч точок, що безпосередньо примикають до деякої точки x . Ціль полягає в знаходженні таких сусідів, тобто. $neighbours(x)$, до будь-яких n точок яких не можна досягти мети оптимально. Ми домагаємося цього шляхом порівняння двох шляхів: p , який починається точкою $p(x)$, відвідує x і закінчується n та іншим шляхом p' , який також починається з $p(x)$, відвідує x і закінчується n , але не містять x . Крім того, кожна точка, що міститься в p або p' , повинна відноситися до $neighbours(x)$.

Існують два випадки, залежно від того, який перехід до x походить з $p(x)$: прямий хід або діагональний. Варто врахувати, якщо x є початком $p(x)$, то $p(x)$ порожньо і відсікання не відбувається.

Прямі переходи: відсікаються будь-які точки $n \in neighbours(s)$, які задовольняють наступному твердженню, зображеному на рисунку 5.5:

$$len((p(x), \dots, n) \setminus x) \leq len(p(x), x, n)$$

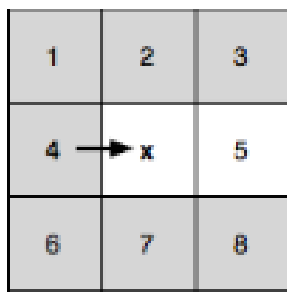


Рисунок 5.5 – Обчислення прямого переходу

Тут $p(x) = 4$ ми відсікаємо всіх сусідів крім $n = 5$.

Діагональні переходи: Тут відмінності в тому, що шлях, який виключає x , має бути строго домінуючим, як зображено на рисунку 5.6:

$$\text{len}((p(x), \dots, n) \setminus x) < \text{len}(p(x), x, n)$$

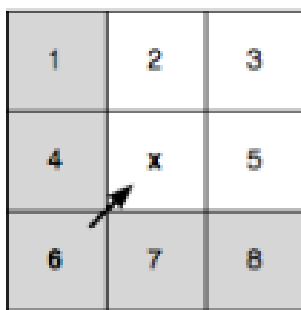


Рисунок 5.6 – Діагонального переходу

Тут $p(x) = 6$ і відсікаються усі сусіди, крім $n = 2$, $n = 3$ та $n = 5$.

Припускаючи, що $neighbours(x)$ не містять перешкод, будемо посылатися на точки, які залишаються після прямого або діагонального відсікання (при необхідності), як природні сусіди x . Вони відповідають не сірим точкам на рисунках 5.7 та 5.8. Коли $neighbours(x)$ містять перешкоди, не можна відсікнути всіх неприродних сусідів. І тут такий сусід вважається вимушеним (штучним).

Точка є вимушеною, якщо

1. n штучний сусід x

2. $\text{len}((p(x), \dots, n) \setminus x) < \text{len}(p(x), x, n)$

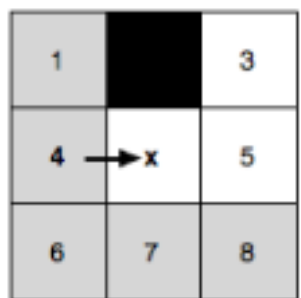


Рисунок 5.7 – Прямий перехід, де $n = 3$ — вимушений.

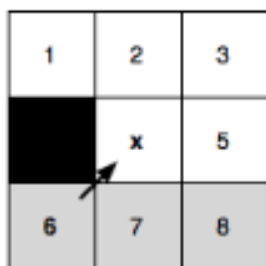


Рисунок 5.8 – Діагонального переміщення; тут $n = 1$ – вимушений сусід.

5.2.2 Опис алгоритму

Точка y є точкою стрибка точки x , y напрямку d , якщо y мінімізує значення k так, що $y = x + kd$, і виконується одна з наступних умов:

Точка y – точка призначення.

У точки y є хоча один сусід, який є вимушеним за визначенням 1.

d – рух по діагоналі та існує точка $z = y + k_i d_i$, яка лежить у k_i кроках у напрямку $d_i \in \{d1, d2\}$, таких що z – точка стрибка з y за умови 1 або 2. Приклад використання наведено на рисунку 5.9.

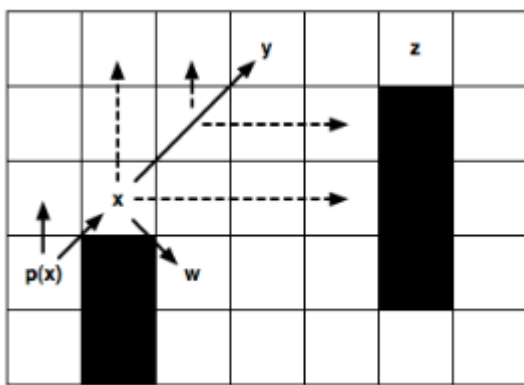


Рисунок 5.9 – Приклад використання алгоритма

Цей рисунок показує приклад точки стрибка, яка визначена умовою 3. Тут ми починаємо в точці x і закінчуємо рух діагоналі, поки не натрапимо на точку y . З y в точку z можна потрапити з кроками по горизонталі. Таким чином, z є наступником точки для стрибка x (за умовою 2), а це y свою чергу визначає y як наступник для

стрибка точки x .

На рисунку 5.10 зображено псевдокод алгоритму визначення наступника:

Задамо: x – поточна точка, s – початок, g – ціль

```

1.  $successors(x) \leftarrow \emptyset$ 
2.  $neighbours(x) \leftarrow prune(x, neighbours(x))$ 
3. for all  $n \in neighbours(x)$  do
4.    $n \leftarrow jump(x, direction(x, n), s, g)$ 
5.   add  $n$  to  $successors(x)$ 
6. return  $successors(x)$ 

```

Рисунок 5.10 – Псевдокод алгоритму визначення наступника

Алгоритм показує, як шукати наступника для поточної точки. Спочатку обрізається безліч сусідів, які безпосередньо примикають до поточної точки x (рядок 2). Тоді замість додавання кожного сусіда n до множини $successors$ (наступників) для x , спробуємо "перестрибнути" до точки, яка знаходиться далі, але яка лежить щодо напрямку x до n (рядки 3-5). Наприклад, якщо ребро $(x; n)$ являє собою рух прямо вправо від x , то дивимося точку стрибка безпосередньо праворуч від x . Якщо така точка, вона додається в набір наступників замість n . Якщо до точки стрибка дійти не виходить, нічого не додається. Процес триває до тих пір, поки всі сусіди не закінчаться, а потім алгоритм поверне список усіх наступників для x (рядок 6).

На рисунку 5.11 зображено псевдокод алгоритму функції стрибка.

Задамо: x – точка звіту, d – напрямок, s – початок, g – ціль

```

1.  $n \leftarrow \text{step}(x, d)$ 
2. if  $n$  – препятствие или находится вне графа then
3.   return null
4. if  $n = g$  then
5.   return  $n$ 
6. if  $\exists n' \in \text{neighbours}(n)$  такие что  $n'$  – принуждённый then
7.   return  $n$ 
8. if  $d$  – диагональное then
9.   for all  $i \in \{1, 2\}$  do
10.    if  $\text{jump}(n, di, s, g)$  не является null then
11.      return  $n$ 
12. return  $\text{jump}(n, d, s, g)$ 

```

Рисунок 5.11 – Псевдокод алгоритма функції стрибка

Для того, щоб знайти окремих наступників для точки стрибка, скористаємося алгоритмом 2. Він вимагає точки звіту x , напрямку руху d , а також початкову точку s і цільову точку g . Алгоритм намагається встановити, чи має x точку для стрибка серед наступників, переміщаючись у напрямку d (рядок 1) і перевіряє, чи задовольняє точка n Визначення 2. У цьому випадку, n позначається точкою стрибка і повертається (рядки 5, 7 і 11). Якщо n не є точкою стрибка, алгоритм рекурсивно повторюється і знову рухається у напрямку d , але цього разу n – нова точка звіту (рядок 12). Рекурсія припиняється, коли зустрічається перешкода і ніякі подальші дії не можуть бути (рядок 3).

Варто звернути увагу, що перед кожним діагональним кроком алгоритм повинен виявити точки стрибка за прямими напрямками (рядки 9-11). Ця перевірка відповідає третій умові визначення 2 і має важливе значення для збереження оптимальності алгоритму.

6 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Тема дипломної роботи є мікроконтролерна система керування автономним пристроєм.

Задачі навігації полягають у забезпеченні точного, своєчасного та безпечного керування, виконанні необхідних розрахунків для маневрування.

До навігації також відносять засоби визначення координат, вимір напрямку та відстані, шляхи вибору та відображення маршруту на карті, вирахування шляху об'єкта, визначення його положення в просторі, оцінка похибки навігаційних приладів.

У навігації використовуються геотехнічні, радіотехнічні, світлотехнічні, астрономічні та інші засоби. Головний елемент перетворення сонячного випромінювання в електрику – це, звичайно ж, матеріал, який, будучи освітленим, перетворює потік світла в електроенергію.

Для подальшого розрахунку обраний обсяг виробництва нового виробу буде складати 300 виробів на рік.

Основні техніко-економічні показники приведені в таблиці 6.1.

Таблиця 6.1 – Основні техніко-економічні показники.

Параметри і показники	Значення
Напруга живлення, В	12 В
Raspberry Pi	Raspberry Pi Model 3A+
Елементна база	Пасивні компоненти/мікросхеми
Маса приладу, г	До 1200
Ціна підприємства	6579.49
Ціна реалізації	6985.34
Поланований об'єм виробництва, шт	300
Точка беззбитковості, шт	92

6.1 Розрахунок собівартості одиниці продукції

Розрахунок витрат на основні матеріали і напівфабрикати власного виробництва.

Витрати на основні матеріали (C_m) і напівфабрикати власного виробництва ($C_{нф}$) визначаються за формулами:

$$C_m = K_m \cdot \sum (H_{vi} \cdot C_i - H_{відх} \cdot C_{відх}) \quad (6.1)$$

де K_m – коефіцієнт, що враховує транспортно-заготовчі витрати на матеріали ($K_m = 1,1$);

H_{vi} – норма витрат на одиницю виробу матеріалу i -го виду (в прийнятих одиницях вимірювання);

C_i – ціна одиниці i -го виду матеріалу, грн. Приймаються реальні або світові ціни на матеріали;

$H_{відх}$ – вага зворотних відходів i -того виду;

$C_{відх}$ – ціна зворотних відходів i -того виду;

Результати розрахунків зводяться в таблицю 6.2.

Таблиця 6.2 – Розрахунок витрат на основні матеріали

Найменування матеріалів	Одиниці виміру	Норма витрат	Ціна за одиницю матеріалу, грн.	Сума, грн
Припій	кг	0,1	200	20
Каніфоль	кг	0,1	225	22,5
Разом вартість основних матеріалів, грн.				42,5
Транспортно-заготівельні витрати, грн..				4,25
Разом вартість основних матеріалів з урахуванням транспортно– заготівельних витрат за винятком відходів, грн.				46,75

Розрахунок витрат на куповані комплектуючі та напівфабрикати виконується за формулою (6.2) і вносяться до таблиці 6.3:

$$C_{\text{куп}} = K_m \cdot \sum (C_{nj} - H_{nj}) \quad (6.2)$$

де K_m – коефіцієнт, що враховує транспортно-заготівельні витрати на куповані комплектуючі ($K_m = 1,1$);

C_{nj} – ціна одиниці i -того виду купованих комплектуючих, грн.;

H_{nj} – кількість купованих комплектуючих i -того виду на виріб, шт.;

Результати розрахунків зводяться в таблицю 6.3

Таблиця 6.3 – Розрахунок витрат на куповані комплектуючі

Найменування купувальних комплектуючих	Кількість куплених виробів, шт.	Ціна за одиницю виробу, грн.	Сума, грн.
Raspberry Pi Model 3A+	1.00	2775.00	2775.00
A4988	2.00	51.00	102.00
MPU6050	1.00	67.00	67.00
Акумулятор 12В 2.3Ач MSU	1.00	533.00	533.00
SY35ST26-0604A	2.00	230.25	460.50
NRF24L01+	1.00	79.25	79.25
Інші			200.00
Разом вартість основних матеріалів			579.49
Транспортно-заготівельні витрати			57.00
Разом вартість основних матеріалів з урахуванням транспортно-заготівельних витрат			4853.24

6.2 Розрахунок основної заробітної платні

Основна заробітна плата визначається як:

$$Zo = Zn + D \quad (6.3)$$

де: $C_{г1}$ – годинна тарифна ставка першого розряду, грн.;

K_i – тарифний коефіцієнт середнього розряду i -того виду робіт;

t_i – трудомісткість i -того виду робіт на один виріб;

Годинна ставка 1-го розряду встановлюється виходячи з мінімальної місячної зарплати та Генеральної Угоди між КМУ та Загальноукраїнськими об'єднаннями організацій роботодавців та підприємців, профспілками. Згідно цієї угоди розмір тарифної ставки 1-го розряду мусить бути не нижчим ніж 120% мінімальної заробітної плати, що встановлена законом на поточний період в перерахунку на годину.

$$C_{г1} = \frac{Z_{min} \cdot 1.2}{\Phi_{міс}} \quad (6.5)$$

де Z_{min} – мінімальна заробітна плата в Україні на момент, що розглядається;

$\Phi_{міс}$ – місячний фонд робочого часу в годинах (може бути прийнятий 168 годин при 21 робочому дні за місяць).

$$C_{г1} = \frac{1921 \cdot 1.2}{168} = 13,72$$

Оплата праці проводиться за тарифною сіткою, що приведена в таблиці 6.4

Таблиця 6.4 – Тарифна сітка

Тарифний розряд	1	2	3	4	5	6	7	8
Тарифний коефіцієнт	1	1.12	1.35	1.56	1.71	2	2.2	2.4

Розрахунок основної заробітної платні виробничих робітників подається в таблиці 6.5

Таблиця 6.5 – Розрахунок основної заробітної платні виробничих робітників

Найменування операції	Розряд	Трудовісткість, н-г	Годинна тарифна ставка, грн.	Сума заробітної плати, грн.
Розмічальні	2	2	15.3664	30.7328
Свердлувальні	3	2	18.522	37.044
Монтажні	4	4	21.4032	85.6128
Складальні	4	3	21.4032	64.2096
Налагоджувальні	5	6	23.4612	140.7672
Разом основна заробітна плата $З_о$				358.3664

6.2.1 Додаткова заробітна плата

Додаткова заробітна плата ($З_д$) виробничих робітників розраховується пропорційно сумі основної заробітної платні.

$$З_д = З_о \cdot 0.1 = 358.3664 \cdot 0.1 = 35.8366 \text{ грн} \quad (6.6)$$

6.2.2 Єдиний соціальний внесок

Згідно Закону України в проекті прийнято ставку єдиного соціального внеску в розмірі 22% від суми основної ($З_о$) і додаткової ($З_д$) заробітної платні.

$$С_{есв} = (З_о + З_д) \cdot 0.22 = (358.3664 + 35.8366) \cdot 0.22 = 86.7246 \text{ грн} \quad (6.7)$$

6.2.3 Витрати на утримання і експлуатацію обладнання

Ці витрати розподіляються на собівартість пропорційно основній заробітній платні (70 % від $З_0$).

$$В_{уео} = З_0 \cdot 0.7 = 358.3664 \cdot 0.7 = 250.8564 \text{ грн} \quad (6.8)$$

6.2.4 Загальновиробничі витрати

Розподіляються на собівартість пропорційно основній заробітній платні (80% від $З_0$).

$$В_{зв} = З_0 \cdot 0.8 = 358.3664 \cdot 0.8 = 286.6931 \text{ грн} \quad (6.9)$$

6.2.5 Втрати від браку

Втрати від браку продукції прийняти 1% від суми попередніх статей витрат. За розрахунком це становить 12.96354 грн.

6.2.6 Інші витрати

Інші витрати приймають в розмірі до 0.9% від суми попередніх статей витрат. Тобто для даного розрахунку ця величина становить 12,2771 грн.

Сума всіх попередніх статей витрат (з 1 до 10) утворює виробничу собівартість ($С_в$). Вона складає 1321.81 грн.

6.2.7 Адміністративні витрати

Розподіляються на собівартість пропорційно основній заробітній платні (80% від $З_0$).

$$А_в = З_0 \cdot 0.8 = 358.3664 \cdot 0.8 = 286.6931 \text{ грн} \quad (6.10)$$

6.2.8 Витрати на збут

Витрати на збут знаходяться пропорційно сумі виробничої собівартості (2.5 % від *виробничої собівартості*). І це становить 33.04 грн.

6.2.9 Прибуток, П.

В калькуляції собівартості розраховуються також прибуток на один виріб, оптова ціна підприємства, ПДВ та відпускна ціна підприємства (ціна продажу). Прибуток на один виріб можна прийняти 20-25% від повної собівартості, він визначається за формулою:

$$П \leq \frac{20\% \cdot C_{п}}{100} \quad (6.11)$$

де $C_{п}$ – повна собівартість, складається із виробничої собівартості, вона складає 1640.73 грн.

$$П \leq \frac{20\% \cdot 1640.73}{100} = 328.14 \text{ грн}$$

Оптова ціна визначається як:

$$Ц = C_{п} + П = 1640.73 + 328.14 = 1968.87 \text{ грн} \quad (6.12)$$

Податок на додану вартість (ПДВ) — форма вилучення в бюджет частини додаткової вартості, яка створюється на усіх стадіях виробництва і визначається як різниця між вартістю реалізованих товарів, робіт та послуг та вартістю матеріальних витрат, віднесених на витрати виробництва та обігу. Становить 20% від суми попередніх витрат. Податок на додану вартість (ПДВ) на дану продукцію згідно законодавству України визначається:

$$ПДВ = 20\% \cdot Ц = 20\% \cdot 1968.87 = 393.77 \text{ грн} \quad (6.13)$$

Відпускна ціна виробу (ціна продажу) розраховується як:

$$Ц_{пр} = Ц + ПДВ = 1968.87 + 393.87 = 2362.74 \text{ грн} \quad (6.14)$$

Всі розглянуті розрахунки зводяться в таблицю 6.6

Таблиця 6.6 – Розрахунок собівартості і ціни приладу

Найменування статті витрат	Величина витрат за
Сировина та матеріали	46.75
Комплектуючі	4853.24
Основна заробітна плата виробничих робітників	358.36
Додаткова заробітна плати виробничих робітників	35.83
Єдиний соціальний внесок	86.72
Витрати на утримання і експлуатацію обладнання	250.85
Загальновиробничі витрати	286.69
Витрати на брак	12.96
Інші виробничі витрати	12.27
Виробнича собівартість	5943.70
Адміністративні витрати	286.6931
Витрати на збут	33.04
Повна собівартість	6263.43
Прибуток	328.14
Ціна підприємства	6591.57
Податок на подану вартість	393.77
Ціна реалізації	6985.34

6.3 Оптимізація обсягу виробництва та побудова графіку беззбитковості

В дипломній роботі виконується побудова графіка беззбитковості. Для цього розраховуються умовно-змінні ($V_{зм}$) витрати на виріб і умовно-постійні ($V_{пост}$) витрати на річний випуск виробів.

$$V_{зм} = C_m + C_{пк} + 3o + 3d + C_{есв} + 0.7 \cdot V_{уео} + 0.2 \cdot A_v \quad (6.15)$$

де: C_m – витрати на основні матеріали, грн.;

$C_{пк}$ – витрати на закупні комплектуючі, грн.;

$3o$ – основна заробітна плата, грн.;

$3d$ – додаткова заробітна плата, грн.;

$C_{есв}$ – відрахування на соціальні заходи;

$V_{уео}$ – витрати на утримання і експлуатацію обладнання, грн;

A_v – адміністративні витрати;

$$V_{зм} = 5943.70 \text{ грн.}$$

$$V_{пост} = (C_n - V_{зм}) \cdot A_{рф} \quad (6.16)$$

де C_n – повна собівартість виробу, грн.

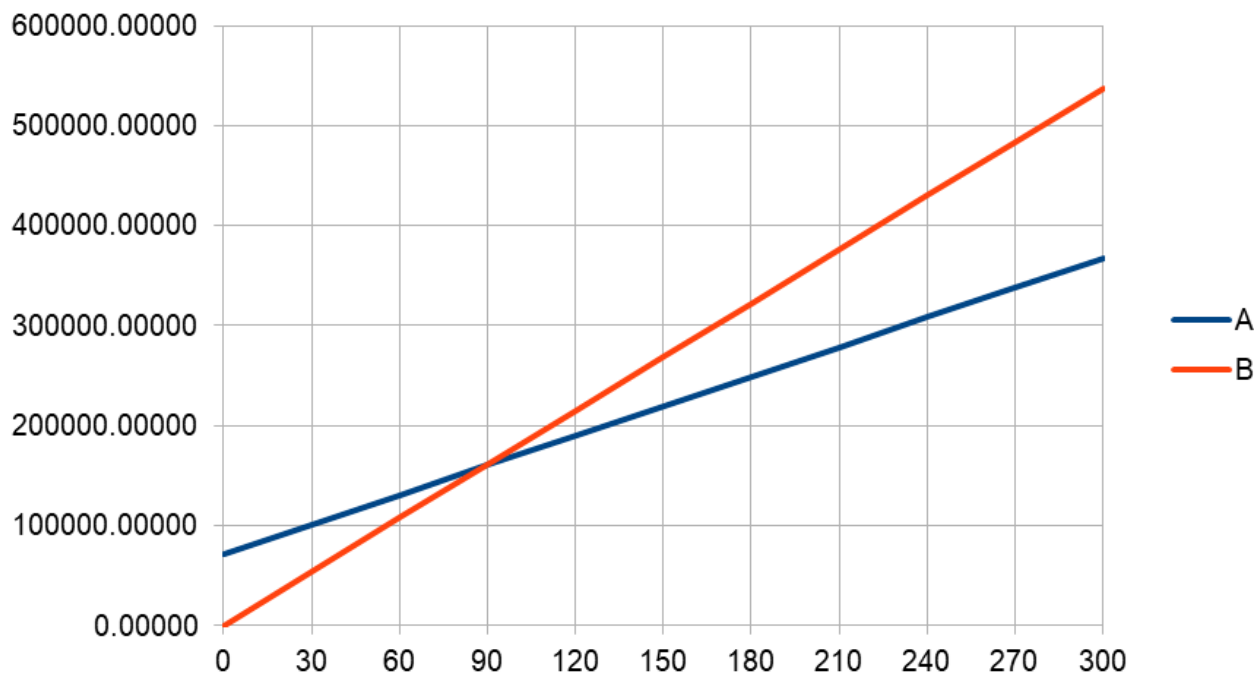
$$V_{пост} = (6263.43 - 5943.70) \cdot 300 = 95919 \text{ грн}$$

Побудова графіка беззбитковості виконується наступним чином. На осі ординат відкладаємо величину $V_{пост}$ від точки $A_{рф}$ на осі абсцис відкладаємо ординату, рівну повній собівартості річного випуску ($A_{рф} C_n$) – точка А. Через точки $V_{пост}$ і А проводимо пряму, яка відображає залежність собівартості від об'єму $C_n = i(A_{рф})$ випуску. Від точки $A_{рф}$ відкладаємо також ординату, рівну об'єму продажів $C_{ф} A_{рф}$ (точка В). Сполучаючи точку початку координат з точкою В,

одержуємо залежність річного об'єму продажів від кількості проданих виробів $V_{\phi} = \Gamma(Ap_{\phi})$. Точка перетину прямих (С) відповідає річному випуску A_{pkr} , при якому відбувається досягнення беззбитковості виробництва.

Розрахуємо критичний об'єм виробництва за формулою:

$$N_{kr} = \frac{B_{\text{пост}}}{\Gamma - B_{\text{зм}}} = \frac{95919}{6985.34 - 5943.70} = 92 \quad (6.17)$$



Риснок 6.1 – Графік беззбитковості

Висновки: В ході розрахунків було встановлено, що, виробнича собівартість приладу складає 5943.70 грн, повна собівартість виробництва складає 6263.43 грн., ціна підприємства складає 6591.57 грн, критичний обсяг виробництва при заданій кількості виробів складає 92 шт.

7 ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

7.1 Загальні питання охорони праці

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів і засобів, що направлені на збереження здоров'я і працездатності людини в процесі праці (Закон України «Про охорону праці» від 22 листопада 2002 року) [13].

Задачею охорони праці є зведення до мінімуму вірогідності уражень або захворювань працюючих з одночасним забезпеченням нормальних умов праці при її максимальній продуктивності, науковий аналіз умов праці, виробничих процесів, а устаткування з погляду можливих аварійних ситуацій, появи небезпечних чинників, виділення шкідливих речовин. На основі такого аналізу визначаються небезпечні ділянки виробництва, можливі аварійні ситуації, розробляються заходи щодо їх запобігання або обмеження наслідків.

7.2 Шкідливі і небезпечні виробничі чинники, їх класифікація

У даному дипломному проекті розглянуті питання забезпечення охорони праці і навколишнього середовища на стадії проектування токарного станка для робочого місця інженера-конструктора.

На робочому місці повинні бути створені певні умови праці, гарантовані Законом України «Про охорону праці».

Під шкідливими виробничими чинниками розуміють чинники, тривалість дії яких на працюючу людину приводить до патологічних змін в організмі людини, або до професійних захворювань. Небезпечні виробничі чинники, це чинники, вплив яких на працюючу людину викликає травму, або летальний результат.

Токарна обробка металів може супроводжуватися наявністю низки шкідливих та небезпечних виробничих факторів, до яких належать:

- електричний струм;
- аерозолі мастильно-охолоджувальної рідини;

- висока температура поверхні оброблюваних деталей та інструментів;
- підвищений рівень вібрації;
- дрібна стружка та шматочки металу, що відлітають.

7.3 Промислова санітарія

Мікроклімат – метеорологічні умови внутрішнього середовища приміщень, які визначаються діючими на людину поєднаннями температури, відносної вологості, швидкості руху повітря і теплового випромінювання.

Параметри мікроклімату відповідно до [15] з урахуванням категорії робіт по енерговитратах для теплого і холодного періодів року приведені в таблиці 7.1, категорія робіт – 1 а. В таблиці 7.2 наведено рівні іонізації повітря приміщень при роботі на ЕОМ.

Таблиця 7.1 – Нормативні параметри мікроклімату для приміщень

Пора року	Категорія робіт згідно з ГОСТ 12.1.005-88	Температура повітря, ° С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
		оптимальна	оптимальна	оптимальна
Холодна	легка – 1 а	22 – 24	60 – 40	0,1
	легка – 1 б	21 – 23	60 – 40	0,1
Тепла	легка – 1 а	23 – 25	60 – 40	0,1
	легка – 1 б	22 – 24	60 – 40	0,2

Таблиця 7.2 – Рівні іонізації повітря приміщень при роботі на ЕОМ

Рівні	Кількість іонів в 1 см ³ повітря	
	n ⁺	n ⁻
Мінімально необхідні	400	600
Оптимальні	1500 – 3000	3000 – 5000
Максимально допустимі	50000	50000

Відповідно до [15] вміст озону у повітрі робочої зони не повинен перевищувати $0,1 \text{ мг/м}^3$; уміст оксидів азоту – 5 мг/м^3 ; вміст пилу – 4 мг/м^3 .

Оптимальні умови досягаються в холодний час центральним опалюванням, а в теплий час вентиляторами і кондиціонерами. Згідно з вимогами до вентиляції, опалення та кондиціювання, мікроклімату приміщення з ЕОМ повинні бути обладнані системами опалення, кондиціювання повітря або припливно-витяжною вентиляцією відповідно до [17].

Природне освітлення на робочому місці представлено одностороннім бічним світлом. Система штучного освітлення є загальною рівномірною.

Розряд зорових робіт – середньої точності, нормовані значення КПО і освітленості на робочих поверхнях приведені в таблиці 7.3.

Таблиця 7.3 – Розряд зорових робіт

Мінімальний розмір об'єкту розрізнення, мм	Розряд зових робіт	Характеристик а типу фону	Контраст об'єкту з фоном	Підрозряд зових робіт	Нормативне значення освітлення	
					Природне е, %	Штучне Е, лк
0,5...1,0	IV	світлий	середній	<i>в</i>	1,5	200

Основним джерелом шуму при роботі за комп'ютером є вентилятори охолодження, які знаходяться у системному блоці ЕОМ. Шум від вентиляторів в системі охолодження комп'ютера постійного рівня, зміна в часі не більше 60 дБА. Захист від шуму проводиться шляхом віддалення системного блоку від працівника і вживанням засобів звукопоглинання [19].

Також існують санітарні норми спектральних показників вібраційного навантаження на оператора, ці норми наведені в таблиці 7.4.

Таблиця 7.4 – Санітарні норми спектральних показників вібраційного навантаження на оператора [21].

Нормативні значення віброприскорення, дБ, в напрямках X_o , Y_o для середньо – геометричних частот смуг, Гц					
2	4	8	16	31,5	63
86	83	83	89	95	101

Монітор комп'ютера є джерелом статичної електрики, оскільки в ньому використовується напруга (14...25 кВ). Захист від статичної електрики припускає вживання спеціальних екранів і заземлення комп'ютера.

Дія електромагнітного випромінювання на людину залежить від напруженості електричного і магнітного полів, частоти коливань, розміру опромінюваної поверхні тіла і індивідуальних особливостей людини.

Вимоги щодо допустимих значень неіонізуючого електромагнітного випромінювання [15]:

1. Напруженість електромагнітного поля на відстані 50 см від джерела по електричній складовій не повинна перевищувати:

- в діапазоні частот 5 кГц – 2 кГц – 25 В/м;
- в діапазоні частот 2 кГц – 400 Гц – 2,5 В/м.

2. Щільність магнітного потоку не повинна перевищувати:

- в діапазоні частот 5 кГц – 2 кГц – 250 нТл;
- в діапазоні частот 2 кГц – 400 Гц – 25 нТл.

3. Поверхневий електростатичний потенціал не повинен перевищувати 500 В.

4. Потужність дози рентгенівського випромінювання на відстані 5 см від екрану і інших поверхонь випромінювання не повинна перевищувати 100 мкР/год [15].

Захисні заходи припускають вживання екранування електромагнітного випромінювання, а також установку поглинаючого екрану між джерелом і

робітником.

7.4 Електробезпека

По ступеню небезпеки поразки людини електричним струмом кімната відноситься до приміщень із підвищеною небезпекою поразки людини електричним струмом, згідно [21], тому що є можливість одночасного доторкання людини до з'єднань із землею металевих конструкцій будинків з однієї сторони та до металевих конструкцій електричного обладнання з іншої сторони. Приймаємо І клас захисту від поразки електричним струмом обслуговуючого персоналу тому, що комп'ютер має робочу ізоляцію й елементи заземлення [17].

Живлення здійснюється змінним струмом від мережі із частотою 50Гц, напругою 220 В. Режим нейтралі – глухо заземлена нейтраль. Захист від ураження електричним струмом – занулення. Споживана потужність комп'ютера – 330Вт.

7.5 Пожежна безпека

Правовою основою діяльності в області пожежної безпеки є Конституція, закон України «Про пожежну безпеку» та інші закони України, ухвалення Верховної Ради України, постанови Кабінету Міністрів і Президента та ін.

Ступінь вогнестійкості приміщення під конструкторське бюро – II, [23] категорія В [22], а за класом приміщення – до II-IIa за ПВЕ

Згідно з [24] приміщення з ЕОМ повинні бути оснащені системою автоматичної пожежної сигналізації. Пожежна автоматика будівель і споруд з димовими пожежними оповісниками і переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 м² площі приміщення з урахуванням гранично допустимої концентрації вогнегасної рідини, відповідно вимогам «Правил пожежної безпеки в Україні».

8.6 Охорона навколишнього середовища

Види матеріальних і енергетичних забруднень, джерелом яких може бути

проектований пристрій є складові елементи літій-іонної батареї, яка використовується для живлення.

Дослідження показують, що у виробництві кіловат-години ємності автомобільних акумуляторів формуються приблизно від 150 до 200 кілограмів еквівалентів двоокису вуглецю. Таким чином, для двох найбільш популярних у наш час електромобілів Nissan Leaf та Тесла Модель S з батареями на 30 кВтг та 100 кВтг вони становлять 5,3 та 17,5 тонн CO₂ відповідно. Це просто неймовірна кількість, для порівняння – викиди в атмосферу авіарейсу зі Стокгольма в Нью-Йорку і назад становлять близько 600 кг CO₂ згідно з розрахунковою моделлю організації ООН ІКАО (Міжнародної організації цивільної авіації). Інше порівняння: у Німеччині щорічні викиди CO₂ на душу населення становлять майже 10 тонн.

Вчені підраховали, що в даний час одному транспортному засобу з двигуном внутрішнього згоряння потрібно 8 років експлуатації, щоб досягти такого ж впливу на навколишнє середовище, як у Тесли. Що стосується Nissan Leaf, то йому для цього знадобиться три роки експлуатації. «Надмірно великі батареї завдають відповідно більше шкоди (клімату). Тому слід подумати, а чи не можна обійтися батареями меншого розміру», каже одна з дослідників Міа Ромаре.

На переробку сировини та виробництво акумуляторних батарей на заводах припадає близько 80% загального впливу на довкілля. Лише 10-20% викидів виникають у результаті видобутку сировини у шахтах.

ВИСНОВКИ

У ході виконання дипломної роботи було виконано розрахунок безпілотного автономного пристрою. Також було написане програмне забезпечення пристрою, яке включає в себе алгоритм пошуку шляху та систему розпізнавання перешкод у реальному часі за допомогою нейронної мережі.

Спроектowana система здатна обробляти зображення з частотою 15FPS тобто данні оновлюються 15 разів на секунду, що, з урахуванням максимальною швидкості приладу в 1.8км\г дозволяє відреагувати на перешкоду, що раптово з'явилась на відстані 3см або більше.

Крім цього в проекті були розглянуті питання економічної доцільності проектування та виготовлення пристрою, та питання охороши праці.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Басов Ю.Г. Светосигнальные устройства. – М.: Транспорт, 1993. – 120 с.
2. Зуев В.Е., Фадеев В.Я. Лазерные навигационные устройства. – М.: Радио и связь, 1987. – 168 с.
3. Таратынов В.П. Судовождение в стесненных районах. – М.: Транспорт, 1980. – 128 с.
4. Афанасьев В.М., Баскин А.С. Лазерные створные маяки // Судовождение и связь. – 1977. – Вып. 7(102). – С. 3–10.
5. <https://www.micropap.com/manuales/SY35ST26-0604A.pdf>
6. <http://autotesla.ru/auto-tesla/model-s/tesla-model-s-texnicheskie-xarakteristiki-i-osobennosti.html>
7. <https://habr.com/ru/post/587580/> [Електронний ресурс]
8. <https://www.youtube.com/watch?v=uCJJFvX7148> [Електронний ресурс]
9. <https://okami.kz/articles/tipy-navigacii-robotov-pylesosov.html> [Електронний ресурс]
10. <https://okami.group/all-about-navigation> [Електронний ресурс]
11. <https://intelpol.ua/rfid-resheniya/rfid-metki/rfid-uhf-metka-dlya-identifikatsii-avtomobilya-anti-falsifikatsionnaya-na-lobovye-steklo> [Електронний ресурс]
12. <https://www.vostok.dp.ua/infa1/rfid/rm/> [Електронний ресурс]
13. Закон України «Про охорону праці» / Законодавство України про охорону праці. – К. : Нова редакція. – 2002.
14. Закон України «Про охорону навколишнього природного середовища». – К. : Україна, 1991. – 59 с. (з усіма редакціями до 2011 року)
15. ДСанПіН 3.3.2-007-98. Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», затверджених постановою Головного державного санітарного лікаря України від 10.12.98 № 7.
16. Правила улаштування електроустановок. ПУЕ – 2014 – Харків.: Форт – 2014.
17. ДСанПіН "Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу". Наказ МОЗ України 08.04.2014 № 248.

18. ДБН В.2.5 – 67-2013. Опалення, вентиляція та кондиціонування. – К. : Мінрегіон України, – 2013.
19. ДБН В.2.5-28-2013 Державні будівельні норми. Інженерне обладнання будинків і споруд. Природне і штучне освітлення. – К. : Мінбуд України, 2006. – 80 с. (зміни 1, 2)
20. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку, інфразвуку. – К., 1999.
21. ДСТУ ГОСТ 12.1.012:2008 ССБТ. Вібраційна безпека. Загальні вимоги. – Введ. 01.07.2008.
22. ДСТУ Б. В.1.1-36:2016. Національний стандарт України. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. – К.: Мінрегіон України, 2016.
23. ДБН В. 1.1-7-02.Державні будівельні норми. Захист від пожежі. Пожежна безпека об'єктів будівництва. – Діє з 01.01.03.
24. НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Наказ Міністерства соціальної політики України від 14.02.2018 № 207.
25. ДСТУ ISO 6826:2015 Двигатели внутреннего сгорания поршневые. Противопожарная защита (ISO 6826:1997, IDT)/ – Введ. 01.01.2016. Приказ от 25.12.2015 № 207
26. ДСТУ 7950:2015 Дизайн і ергономіка. Робоче місце під час виконання робіт стоячи. Загальні ергономічні вимоги. – Діє 01.09.2016
27. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. – діє з 01.07.2017
28. ГСТУ 32.001-94 Выбросы загрязняющих веществ с отработанными газами тепловозных дизелей. Нормы и методы определения. – діє з 01.01.1995