

**Міністерство освіти і науки України**

**Луцький національний технічний університет**

(повне найменування закладу вищої освіти)

**Факультет комп'ютерних та інформаційних технологій**

(повне найменування факультету)

**Кафедра комп'ютерної інженерії та кібербезпеки**

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**МОБІЛЬНИЙ РОБОТ З ЖЕСТОВИМ КЕРУВАННЯМ НА БАЗІ  
ARDUINO UNO**

**A MOBILE ROBOT WITH GESTURE CONTROL BASED ON  
ARDUINO UNO**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи КІ-42

Шинкарук Артем Володимирович

(підпис)

Керівник:

к.т.н., доцент

Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 19 » червня 2024 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Шинкаруку Артему Володимировичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Мобільний робот з жестовим керуванням на базі Arduino UNO*

Керівник роботи *к.т.н., доцент Гринюк Сергій Васильович*

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2024р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

*Вступ*

*Аналіз проблеми за темою роботи та постановка завдань, дослідження існуючих методів та засоби проведення дослідження, розробка програмно-апаратних засобів для керування робототехнічною частиною*

5. Перелік графічного (ілюстративного) матеріалу:

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Гринюк С.В., доцент</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Гринюк С.В., доцент</i>		
<i>Практична реалізація системи керування жестами</i>	<i>Гринюк С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	_____ %		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Проведення огляду літератури із досліджуваної проблеми</i>	до 15.02.2024 р.	Виконано
2.	<i>Проведення аналізу сфер використання мобільних роботів</i>	до 15.03.2024 р.	Виконано
3.	<i>Розробка систем керування мобільним роботом</i>	до 04.05.2024 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 10.05.2024 р.	Виконано
6.	<i>Формування додатків</i>	до 15.05.2024 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 20.05.2024 р.	Виконано
8.	<i>Нормоконтроль</i>	до 01.06.2024 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 04.06.2024 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 11.06.2024 р.	Виконано

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Шинкарук А.В.

\_\_\_\_\_ (прізвище, ініціали)

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Гринюк С.В.

\_\_\_\_\_ (прізвище, ініціали)

## АНОТАЦІЯ

Шинкарук А. В. Мобільний робот з жестовим керуванням на базі Arduino UNO. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 52 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

В першому розділі аналізуються мобільні роботи та способи ними керування.

В другому розділі здійснено вибір та обґрунтування апаратних засобів розробки керування робототехнічною системою за допомогою жестів. Обрано Arduino Uno та відповідні датчики.

Третій розділ присвячено опису середовища програмування та розробки апаратних та програмних засобів для керування мобільним роботом.

Ключові слова: Arduino Uno, радіомодуль, плата сервоприводів, робот.

## ANNOTATION

Shinkaruk A. V. A mobile robot with gesture control based on Arduino UNO. Manuscript.

Bachelor's qualifying thesis of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 52 p.

The qualification work consists of an introduction, three sections, conclusions, a list of used sources, and appendices.

The first chapter analyzes mobile robots and ways of managing them.

In the second section, the selection and justification of the hardware tools for the development of control of the robotic system using gestures is carried out. An Arduino Uno and the appropriate sensors are selected.

The third chapter is devoted to the description of the programming environment and the development of hardware and software tools for controlling a mobile robot.

Keywords: Arduino Uno, radio module, servo board, robot.

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 ТЕОРЕТИЧНА ЧАСТИНА .....	6
1.1 Аналіз сфер використання мобільних роботів .....	6
1.2 Способи керування мобільними роботами .....	13
РОЗДІЛ 2 ВИБІР КОМПОНЕНТІВ ДЛЯ РОБОТИЗОВАНОЇ ПЛАТФОРМИ... ..	19
2.1 Огляд мікроконтролерів Arduino.....	19
2.2 Плата драйвера моторів L298P .....	24
2.3 Модуль GY-521 .....	27
2.4 Модуль NRF24L01 .....	28
РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ КЕРУВАННЯ МОБІЛЬНИМ РОБОТОМ ... ..	31
3.1 Середовище Arduino IDE .....	31
3.2 Апаратна розробка .....	33
3.3 Програмна розробка .....	39
ВИСНОВКИ .....	43
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	44
ДОДАТКИ .....	49

## ВСТУП

Робототехніка – це прикладна наука, яка займається розробкою автоматизованих технічних систем і є найважливішою технічною базою для інтенсифікації виробництва. В даний час велика кількість завдань пов'язана з організацією автоматичного руху робота, виконанням деяких основних дій. Серед різних класів роботів важливим сучасним класом є мобільний робот, який розміщений на мобільній платформі, що використовується в промисловості (робототехніка), екології (динамічний моніторинг навколишнього середовища, робота в екологічно шкідливих умовах), військовій сфері (розвідувальний робот), побутовий робот (автономні пилососи, газонокосарки тощо).

Однак створення роботів, які впевнено рухаються навіть на рівній поверхні, де перешкоди для них нездоланні, важко з кількох причин, включаючи недосконалість алгоритмів управління. Надійний алгоритм управління рухом електроприводів мобільних роботів є критично важливим для успішної роботи.

Сучасні мобільні роботи можуть самостійно переміщатися в навколишньому просторі і виконувати необхідні дії за допомогою маніпуляторів. Робот оснащений системою технічного зору і комплексом інформаційних датчиків, здатних сформуванати комплексне уявлення про поточну ситуацію. База знань робота, дозволяє йому самостійно орієнтуватися в навколишньому середовищі і приймати рішення про дії, необхідних для вирішення поставленого завдання. Таким чином, маніпуляційний мобільний робот являє собою «інтелектуальну» технічну систему, здатну до автономного поведінки. Проте, в більшості завдань, які виконуються в заздалегідь не визначених умовах і пов'язаних з високою «ціною» помилки при невірних діях, як і раніше передбачається участь людини–оператора в управлінні роботом.

Застосування робототехніки в різних сферах, пов'язаних з вирішенням спеціальних завдань, вимагає максимальної спрощення способів взаємодії людини і робота. В якості одного із способів забезпечення комфортної взаємодії з робототехнічною системою є людська рука, яка може бути використана в якості

інтерфейсу вводу. Жести рук є потужним каналом зв'язку, який формує основну частину передачі інформації в нашому повсякденному житті. У порівнянні з традиційними пристроями людиномашинного взаємодії, жести є менш нав'язливим, більш простим та зручним і природним способом взаємодії для користувачів. Проте, виразність жестів все ще залишається недостатньо вивченою для вирішення проблеми людиномашинного взаємодії

Метою роботи є розробка та удосконалення системи управління мобільним роботом на базі платформи Arduino.

Об'єкт дослідження – мобільний робот, керований жестами, який використовує платформу Arduino UNO для обробки даних та управління рухом.

Предмет дослідження – методи та алгоритми керування мобільним роботом.

Завдання, які необхідно виконати :

- Аналіз використання мобільних роботів та систем керування ними.
- Вибір відповідних апаратних компонентів та датчиків робототехнічної системи.
- Монтаж та налаштування апаратних компонентів і розробка програмного коду для обробки даних з сенсорів для управління роботом.

## РОЗДІЛ 1

### ТЕОРЕТИЧНА ЧАСТИНА

#### 1.1 Аналіз сфер використання мобільних роботів

Мобільний робот – це автоматична машина, яка здатна пересуватися. Мобільна робототехніка зазвичай вважається підгалуззю робототехніки та інформаційної інженерії [1].

Мобільні роботи мають можливість пересуватися в навколишньому середовищі й не прикріплені до одного фізичного місця. Мобільні роботи можуть бути «автономними», що означає, що вони здатні орієнтуватися в неконтрольованому середовищі без потреби у фізичних або електромеханічних пристроях наведення. Крім того, мобільні роботи можуть покладатися на пристрої навігації, які дозволяють їм подорожувати заздалегідь визначеним навігаційним маршрутом у відносно контрольованому просторі. Навпаки, промислові роботи зазвичай більш-менш стаціонарні, складаються із з'єднаної руки (багатоланкового маніпулятора) і вузла захоплення (або кінцевого ефектора), прикріпленого до нерухокої поверхні.

Мобільні роботи стали більш звичним явищем у комерційних і промислових умовах. Лікарні вже багато років використовують автономних мобільних роботів для переміщення матеріалів. На складах встановлено мобільні роботизовані системи для ефективного переміщення матеріалів із складських полиць до зон виконання замовлень. Мобільні роботи також є основною темою поточних досліджень, і майже в кожному великому університеті є одна або кілька лабораторій, які зосереджуються на дослідженні мобільних роботів. Мобільні роботи також зустрічаються в промислових, військових і охоронних установах.

Компонентами мобільного робота є контролер, датчики, виконавчі механізми та система живлення. Контролер, як правило, є мікропроцесором, вбудованим мікроконтролером або персональним комп'ютером (ПК). Використовувані датчики залежать від вимог робота. Вимогами можуть бути

розрахунок мертвої точки, тактильне визначення та визначення наближення, триангуляційне визначення дальності, уникнення зіткнень, визначення місця розташування та інші спеціальні програми. Приводи зазвичай відносяться до двигунів, які рухають робота на колесах або ногах. Для живлення мобільного робота ми зазвичай використовуємо джерело постійного струму (тобто акумулятор) замість змінного [2].

З кожним роком робототехніка розвивається все більше і більше. Створюються нові підходи до вирішення проблем руху, локалізації, автоматизації роботів. Багато моделей досягають чималих успіхів у вирішенні різних завдань. Чимало технічних комплексів створюється для військових цілей: виявлення мети, її ліквідації. Створюються роботи пожежники; роботи рятувальники, здатні діставати людей з води, з уламків повалених будівель.

Процес роботизації торкнувся і таку специфічну область як забезпечення громадської безпеки: ось вже більше 20 років в арсеналі спецслужб і поліцейських підрозділів знаходяться мобільні роботи робототехнічні комплекси. До сих пір немає чіткого уявлення про те, яку машину можна вважати роботом. В енциклопедичному словнику роботом називається автоматична система (машина), оснащена датчиками, що сприймають інформацію про навколишньому середовищу, і виконавчими механізмами, здатна за допомогою блоку управління цілеспрямовано вести себе в мінливих обстановці. Характерною особливістю робота вважається здатність частково або повністю виконувати рухові і інтелектуальні функції людини. Від звичайної автоматичної системи (наприклад, верстата-автомата) робот відрізняється багатоцільовим призначенням, більшою універсальністю, можливістю перебудови на виконання різноманітних функцій. На практиці ж поняття «робот» поширюють і на будь-які дистанційно керовані транспортні кошти, забезпечені системою очуєтвленія (як мінімум, системою технічного зору). Робот покликаний замінити людини в випадках, коли виконання завдання знаходиться за межами людських можливостей або пов'язане з надмірної загрозою здоров'ю та життю людини, а

також при нестачі професійно підготовленого персоналу для виконання трудомістких і циклічно повторюваних завдань [3].

Однією з багатьох тенденцій в робототехніці є перехід від телекерованих систем, які вимагають постійної участі людини для виконання всіх дій робота до автономних систем, в яких оператор лише вказує кінцеві і проміжні цілі. Це зручно для проведення інопланетних досліджень, де велика затримка сигналу не дозволяє здійснювати телеуправління.

У багатьох країнах, робототехніка є вже окремою дисципліною, що вивчається в навчальних закладах, і не тільки в профільних вузах, але і в школах, і, відповідно, в педагогічних вузах. Попит на фахівців, що володіють знаннями в цій області, щорічно зростає. Це обумовлено багатьма об'єктивними факторами, вирішальним з яких, звичайно ж, є бурхливий розвиток робототехніки.

Особливе значення, так само займає розвиток так званої, автономної і мобільної робототехніки. В силу того, що робототехніка часто застосовується для виконання робіт в недоступних людині місцях, то саме мобільна робототехніка часто є пріоритетом. Однак, не тільки в перерахованих вище випадках потрібно мобільна робототехніка, певною гілкою її розвитку є автономна робототехніка, що передбачає практично повне виключення втручання людської діяльності для виконання будь-яких завдань. Іншими словами, така робототехніка є найбільш перспективною і цікавою в плані розробки.

Для здійснення автономності, в мобільній робототехніці, все частіше застосовується штучний інтелект, а точніше його елементи. З розвитком хмарних технологій, варто звернути увагу, на інтеграцію і об'єднання різних по потужності обчислювальних потужностей і сервісів штучного інтелекту, для повномасштабної реалізації повної мобілізації та автономності робототехніки [3].

Можна виділити два основні напрями розвитку мобільних роботів.

Перший напрям – це послідовне створення більш складних повністю автономних систем для умов з різною мірою визначеності і важчих прикладних

завдань. Ускладнення умов в основному визначається мірою неорганізованості зовнішнього середовища і наявністю апріорної інформації про неї. Стосовно середовища це може бути робота в приміщеннях, рух по дорогах, рух по непідготовленій місцевості. Міра невизначеності наростає в такій послідовності: наявність детального плану (карти) з позначенням усіх прохідних і заборонених зон (перешкод), наявність карти з вказівкою мережі доріг і великих заборонених для руху зон (район щільної міської забудови, гора, ліс, річка, болото) без вказівки локальних стаціонарних або рухливих перешкод і, нарешті, відсутність якої-небудь достовірної карти або плану. Остання ситуація можлива не лише при висадці на незнайомій планеті, але і на Землі в районах великомасштабних техногенних або природних катастроф. Ще ширше може варіюватися складність вирішуваних завдань від простого руху в заданому районі до активного перетворення середовища по трасі руху (проробляє проходів в завалах, прокладення ліній зв'язку і трубопроводів, риття траншей, наведення переправ, створення земляних захисних споруд і т. д. Перший напрям в основному використовується при розробці промислових роботів, що функціонують в середовищах з високою мірою визначеності [4].

Другий напрям – це створення інтегрованих людино-машинних систем з послідовним збільшенням числа функцій, що виконуються машиною. Звільнення оператора від рутинних операцій і виключення його з безпосереднього управління різко скорочує об'єм передаючої через ефір інформації, а можливість втручання оператора в складних ситуаціях розширює коло вирішуваних завдань. Інтелектуальні можливості комбінованої системи визначаються інтелектом і професійною підготовкою оператора, а її невтомність, надійність і безпомилковість при виконанні простих операцій автоматом. Як вважають зарубіжні фахівці, другий напрям особливо привабливий для військових у зв'язку з тим, що на сьогодні вже створене достатнє число прийнятно працюючих зразків дистанційно керованих комплексів на базі штатного озброєння. У перспективі людино- машинні системи можуть досягти такого рівня розвитку, коли роль людини зведеться лише до постановки завдань,

а це означає, що обидва напрями приведуть до однакового результату, оскільки будь-який «автономний» робот повинен отримати завдання, а потім доповісти про його виконання або повідомити про неможливість виконання. Трирівнева структура системи управління автономного робота у разі потреби дозволяє органічно використовувати і різні режими дистанційного керування. Операторові надається можливість на стратегічному рівні управляти «макродіями», реалізуючи супервізорне управління, або узяти на себе додатково функції тактичного рівня, задаючи напрям і швидкість руху. У випадках коли використовується режим дистанційного керування, на автоматичну систему доцільно покласти рішення допоміжних завдань. Наприклад, таких, як прокладення маршруту по електронно-цифровій карті, відображення на цій карті місцезнаходження робота (автоштурман), відображення на екрані тривимірної моделі спостережуваної сцени і поєднання її з монотелевізійним зображенням, попередження про можливе виникнення нештатних ситуацій при помилкових діях оператора, фіксація усіх його дій в електронному протоколі і т.д. [4] .

Дуже зручною і перспективною експериментальною платформою для, відпрацювання навичок створення мобільної робототехніки, є мікроконтролерна платформа ARDUINO. Застосування можливостей мобільних робототехнічних комплексів на основі ARDUINO допомагає удосконалити навички з дисциплін математики, інформатики, відпрацювати професійні навички відразу по декількох суміжних дисциплін: механіка, теорія управління, програмування, теорія інформації. А використання датчиків допоможе вибудувати міжпредметні зв'язки з фізикою, біологією і хімією. Попит з комплексних знань сприяє розвитку комунікативних навичок між творчими командами .

Зручність ARDUINO в якості стартової моделі для створення простої мобільної робототехніки, обумовлюється великою кількістю інформації і просто величезного числа доступних комплектів розширень (датчиків, двигунів, модулів бездротового зв'язку і т.д.). Не складає ніяких труднощів реалізувати на цій платформі найпростішого мобільного робота, керованого за допомогою

смартфона або ж пульта управління. Однак в залежності від типу проекту, буде підніматися питання про час автономної роботи. Саме тут можна побачити всі переваги платформи ARDUINO, яка має величезне число комплектів розширень, комбінуючи які можливо створити нове, нестандартне рішення щодо забезпечення всієї мобільної системи енергією в потрібній кількості на потрібний час.

Роботів створюють для того, щоб вони могли замінити людини в складних для нього робочих умовах.

В кінці лютого 2020 року китайська рітейлери стали масово запускати роботів для доставки замовлень, щоб запобігти поширенню смертоносного коронавірусу [5].

Компанія Meituan Dianping, яка спочатку виступила з ініціативою «безконтактної доставки» по Китаю, вже почала використовувати автономні транспортні засоби для поставок продуктів в районі Шуньї в Пекіні і планує запуснути аналогічні служби доставки роботів в інших районах столиці [5].

Компанія почала тестувати роботів і безпілотників для доставки в 2019 році, але це перший випадок розгортання автономних засобів доставки на дорогах загального користування. Автономний робот здатний перевозити до 100 кг товарів і доставляти від трьох до п'яти замовлень за кожну поїздку.

У серпні 2019 року Huawei оприлюднила дослідження Global Industry Vision (GIV), в якому прогнозує шляху розвитку технологій і галузей [6]. У доповіді перераховані чотири типи роботів, які, на думку компанії, можуть стати членами сім'ї.

Роботи-дворецькі можуть виконувати такі побутові завдання, як складання одягу, прибирання пилососом і приготування кави. Такі роботи в різних формах використовуються в 200 млн будинках по всьому світу до серпня 2019 року, повідомили в Huawei [6].

Роботи-компаньйони можуть ділитися на три категорії в залежності від завдань: навчання, лікування і дружба. Наприклад, такі роботи почали використовуватися для взаємодії з дітьми, хворими на аутизм.

Роботи-доглядальниці можуть постійно перебувати з маломобільними пацієнтами та оперативно сповіщати родичів і лікарню про те, якщо у хворого стався серцевий напад. Для цього роботи збирають дані з датчиків, встановлених на людину і в різних частинах будинку.

Біонічні роботи будуть представлені в трьох формах: екзоскелети, протези і додаткові пристосування. Згідно з прогнозами Huawei, обсяг світового ринку екзоскелетів до 2025 року досягнуть \$ 2 млрд [6]. До цього часу роботи все частіше будуть використовуватися і в промисловому секторі: на 10 тис. співробітників припадатиме 103 робота.

Інтелектуальні засоби автоматизації, які вже трансформували багато галузей промисловості, вирішуватимуть більш небезпечні, повторювані і високоточні завдання на благо людини, підвищуючи безпеку і продуктивність.

Сучасний етап розвитку суспільства характеризується прискореними темпами освоєння техніки і технологій, і розвиток автоматизованих систем є в ньому закономірним процесом. Це викликає необхідність вміння вирішувати завдання за допомогою роботів і автоматів, які можна спроектувати в реальній моделі, тобто безпосередньо сконструювати і запрограмувати.

Кожне наступне покоління роботів володіє великими можливостями і досконалістю, але не виключає попереднього; вони взаємно доповнюють один одного і знаходять застосування відповідно до своїх функціональних можливостях і умов економічної доцільності. До теперішнього часу сформувалося три покоління роботів [7]:

- роботи першого покоління (з програмним управлінням), застосовують для: обслуговування верстатів, пресів, печей, зварювальних установок і машин; виконання основних технологічних процесів (різання, збірки, зварювання); вантажно-розвантажувальних і складських робіт;

- роботи другого покоління відрізняються від роботів першого наявністю чутливих пристроїв (дотик, телевізійне зір), мають більш складне управляючий пристрій;

– роботи третього покоління (інтегральні роботи) на відміну від роботів другого покоління обробляють інформацію, що отримується від органів почуттів. Ці роботи застосовують для робіт, що вимагають розпізнавання образів (робота за кресленням), а також протікають в складних і мінливих умовах.

До мобільному роботу пред'являються наступні загальні вимоги:

– роботу необхідно мати високі рухливість і прохідність в міських умовах, всередині будівель і споруд, в зонах руйнувань, на пересіченій місцевості, як на твердих гладких покриттях, так і на деформованих ґрунтових підставах;

– робот повинен надійно діяти як в непередбачених природних умовах так і в середовищі, спеціально пристосованій для проживання людини (всередині будинків, в транспортних комунікаціях), вписуватися в міські транспортні потоки або рухатися в складі транспортних колон;

– конструкція робота повинна забезпечувати швидке розгортання при виконанні спецоперацій і його високу мобільність.

## **1.2 Способи керування мобільними роботами**

Однією з фундаментальних проблем мобільної робототехніки є автономне переміщення. Для виконання цього завдання робот повинен знати своє положення щодо становища своєї мети. Для ефективного функціонування інтелектуальні роботи забезпечені системою сприйняття довкілля, засобами аналізу ситуацій та здійснюють планування руху. При цьому виключно важливою є теоретико-механічна складова задачі, що включає дослідження загальних глобальних властивостей керованої системи.

Пристрій управління робота здійснює автоматичне управління його виконавськими системами – маніпуляційними і пересування, утворюючи в сукупності з ними як об'єктами управління систему автоматичного управління робота. Крім того, пристрої управління роботів часто використовують і для управління різними іншими об'єктами (технологічним устаткуванням,

транспортними пристроями і т. п.), які працюють спільно з роботом, утворюючи з ним єдиний технологічний комплекс [8].

Як вже було відмічено, за способом управління розрізняють наступні системи управління роботів і відповідні пристрої управління:

- програмні пристрої, в яких управління здійснюється за попередньо складеною незмінною в процесі реалізації програмою;
- адаптивні пристрої, в яких управління здійснюється залежно від інформації про поточний стан зовнішнього середовища і самого робота, отриманою в процесі управління від сенсорних пристроїв;
- інтелектуальні пристрої, в яких для адаптації і виконання інших функцій робота використовуються методи штучного інтелекту.

Пристрої управління можуть бути індивідуальними, такими, що входять до складу кожного робота, і груповими, керівниками декількома роботами. Конструктивно індивідуальні пристрої управління виконують зазвичай окремо від механічної частини робота. Переважна більшість роботів мають електронні пристрої управління, виконані на мікропроцесорній базі. Проте існують і неелектричні пристрої управління роботів, що найчастіше реалізуються на пневматиці і призначені для застосування в особливих вибухо та жаронебезпечних умовах.

У робототехніці існують 3 завдання управління: управління маніпуляторами, системою пересування і спільне управління групою роботів. Використані для вирішення цих завдань способи управління можна класифікувати наступним чином.

По мірі участі людини в процесі управління існують системи:

- автоматичного;
- автоматизованого;
- ручного управління.

За типом алгоритму автоматичного управління розрізняють системи:

- програмного;
- адаптивного;

- інтелектуального управління.

За типом руху існують системи управління :

- безперервні (контурні);
- дискретні позиційні (кроками «від точки до точки»);
- дискретні циклові (з одним кроком по кожній координаті).

По виду керованих змінних розрізняють системи управління:

- положенням (позицією);
- швидкістю;
- силою (моментом).

Стосовно управління маніпуляторами ці способи означають управління абсолютними координатами його робочого органу, швидкістю і силою.

Часто перераховані способи управління застосовуються в комбінації одночасно по різних координатах, у вигляді деякої функціональної залежності однієї керованої змінної від іншої або шляхом послідовного переходу від одного способу управління до іншого.

Система управління мобільним роботом в загальному випадку виконує наступні функції [9]:

- сприйняття і розпізнавання інформації, що поступає із зовнішнього світу від сенсорів;
- спілкування з людиною;
- створення і коригування моделі середовища шляхом навчання в процесі спілкування з людиною, сприйняття сигналів з датчиків і виконання дій;
- планування і перепланування поведінки;
- управління виконанням дій;
- управління приводами;
- спілкування з іншими роботами.

Розглянемо ці особливості стосовно мобільних роботів, що пересуваються по довільній місцевості. В цьому випадку модель зовнішнього середовища є картою місцевості, яка окрім апріорних відомостей складається і уточнюється в ході руху на основі сенсорної інформації. Ця модель повинна мати як мінімум

два рівні по масштабу: перший в межах відстані, що досягається сенсорними системами, другий – для місцевості безпосередньо перед роботом.

На рисунку 1.1 приведена типова схема системи управління рухом мобільного робота. Блок управління рухом здійснює управління тяговими приводами і приводами повороту шасі. Блок побудови маршруту синтезує траєкторію руху, оптимізуючи її, по мінімуму витрати енергії, що особливо важливо для роботів з автономним енергоживленням, або по мінімуму часу виконання завдання, коли в постановку завдання входить умова забезпечення максимальної швидкодії.

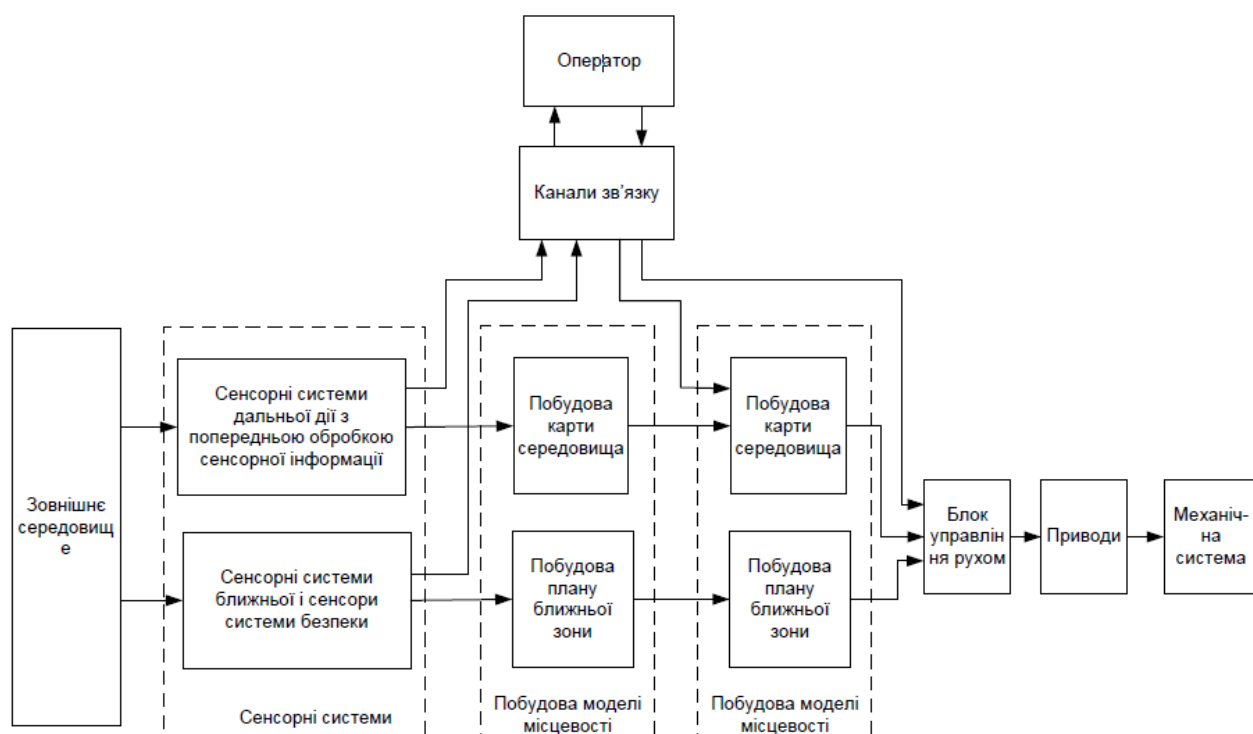


Рисунок 1.1 – Функціональна схема системи управління рухом по місцевості мобільного робота

Блок побудови карти (плану) місцевості представляє її у формі, зручній для вирішення завдання вибору маршруту, зокрема, з виділенням непереборних перешкод і небезпечних або неясних ділянок. Якщо виконання завдання вимагає знання більшої ділянки місцевості, чим дають сенсорні системи в початковому положенні робота, тобто до початку руху, карта місцевості формується і

передається у блок побудови маршруту фрагментами у міру пересування робота. Такими ж ділянками послідовно здійснюється і синтез траєкторії руху. В цьому випадку перше наближення для усієї траєкторії робиться за наявною апріорною інформацією про місцевість або, як мінімум, визначається загальний напрям руху.

На нижньому загальносистемному рівні системи управління знаходиться канал управління рухом у ближній зоні, що включає найбільш детальну модель цієї зони і що реалізовує алгоритми забезпечення безпеки руху.

Необхідний рівень адаптації і штучного інтелекту системи визначається мірою невизначеності і складності місцевості і характером що підлягають виконанню завдань, тобто призначенням робота. В останньому відношенні основне значення має мінімально необхідна міра автономності управління роботом, що визначається переліком його дій, які повинні виконуватися без участі людини-оператора. При цьому враховується і можливість тимчасової втрати зв'язку з оператором, обмежена пропускна спроможність каналів зв'язку, неповнота і обмежена достовірність отримуваної оператором від робота інформації і її затримка в часі.

З розглянутих способів управління оператором для управління пересуванням робота не застосовується тільки система копіюючого управління, оскільки вона базується на використанні кінематики маніпуляторів.

Командне управління, в даному випадку, приводами коліс шасі робота застосовується для виконання дискретних команд «вперед-назад», «вліво - управо» з дискретним завданням швидкості руху і команди «стоп». Крім того, таке управління використовується для рухливих, зазвичай двоступеневих, передавальних телевізійних камер і локаторів огляд місцевості, а також різного навісного устаткування, яким може бути оснащено шасі робота.

Управління за допомогою джойстика – сьогодні основний спосіб дистанційного керування (по кабелю або радіоканалу) мобільними роботами.

Інтерактивний режим управління має особливо велике значення при русі мобільних роботів в невідомому середовищі. В цьому випадку діалоговий режим

часто потрібний при плануванні маршруту, а також при виникненні складних і непередбачених перешкод. Як правило, можливість оперативного підключення при цьому людини-оператора особливо важлива.

Системи управління мобільних роботів підводного, повітряного і космічного базування мають в цілому ту ж функціональну структуру і використовують ті ж способи управління, що і наземні роботи. Проте, звичайно, вони відрізняються системами пересування і динамікою.

Основні тенденції розвитку систем управління мобільних роботів з боку людини-оператора наступні:

- розвиток систем відображення зовнішньої ситуації навколо робота, включаючи ефект присутності оператора безпосередньо у робота і візуалізації зовнішнього середовища (віртуальна реальність);
- розвиток інтелектуальних комп'ютерних систем інформаційної підтримки оператора, заснованих на моделюванні і прогнозуванні в реальному часі поведінки робота і довкілля;
- розвиток чисто електронних, позбавлених рухливих механічних частин, засобів управління роботом від оператора.

## РОЗДІЛ 2

### ВИБІР КОМПОНЕНТІВ ДЛЯ РОБОТИЗОВАНОЇ ПЛАТФОРМИ

#### 2.1 Огляд мікроконтролерів Arduino

Arduino – це інструмент для проектування електронних пристроїв, які більш тісно взаємодіють із фізичним середовищем, ніж звичайні персональні комп'ютери, які фактично не виходять за рамки віртуальності. Це платформа, призначена для управління фізичними процесами за допомогою ЕСМ з відкритим кодом, побудована на простій друкованій платі з сучасним середовищем для написання програмного забезпечення.

Плати Arduino побудовані на основі мікроконтролерів Atmel, а також зв'язкових елементів для програмування та інтеграції з іншими схемами. Лінійний стабілізатор напруги +5В або +3,3В розташований на платі. Синхронізація здійснюється на частотах 8, 16 або 87 МГц за допомогою кварцевого генератора.

Мікроконтролер попередньо прошивається завантажувачем, тому зовнішній програматор не потрібен.

Переваги плат сімейства Arduino:

- велика кількість доступних опцій в лінійці Arduino з можливістю вибору найбільш підходящого готового контролера з великого списку пристроїв, які мають широкий діапазон змінних параметрів;
- наявність плат розширення, призначених для збільшення функціональності та виконання конкретних технічних завдань без необхідності самостійного проектування додаткової периферії (плати керування двигуном, плати датчиків, бездротових інтерфейсів, дисплеїв, пристроїв введення) – кілька десятків типів, більше 300 версій;
- середовище програмування, яке повністю адаптоване для кінцевого користувача і підходить для всієї лінійки плат Arduino та їх клонів, включаючи програмне забезпечення для програмування контролерів для Android;

безкоштовна ліцензія на пристрої та програмне забезпечення.

Arduino Uno (рис. 2.1) – це контролер, побудований на ATmega328. Платформа має 14 цифрових входів/виходів (6 з яких можна використовувати як ШІМ-виходи), 6 аналогових входів, кварцевий генератор 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP і кнопку скидання. Для роботи необхідно підключити платформу до комп'ютера через USB-кабель або жити від адаптера AC/DC або батарейок [10].

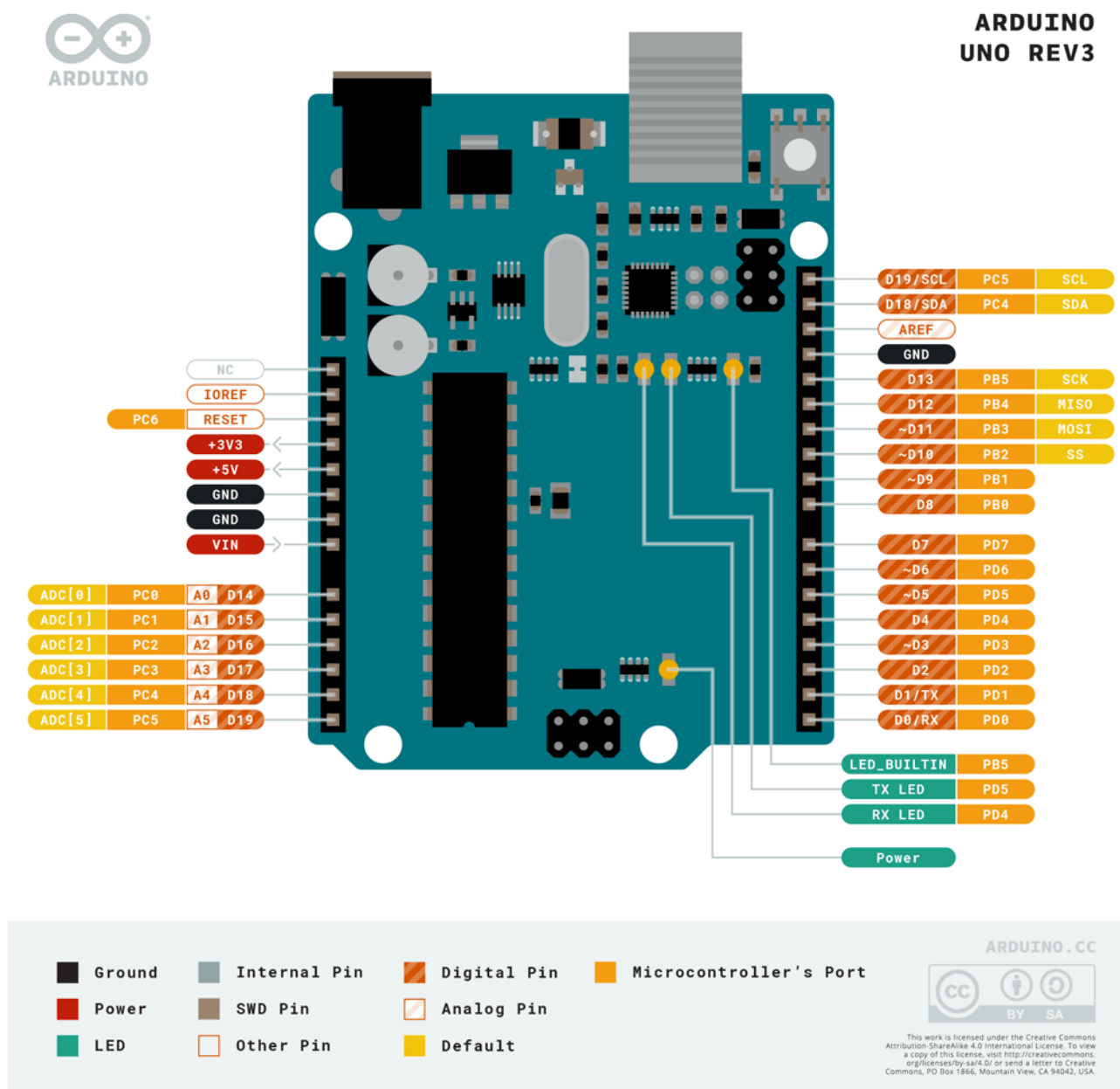


Рисунок 2.1 – Плата Arduino Uno [10]

Завдяки технічній оснащеності платформа Arduino ідеально підходить для навчального процесу проектування різних мехатронних систем і роботів завдяки чіткому середовищу програмування та можливості спостерігати фізичні процеси в реальному часі. Більш потужні плати Arduino (Due) застосовні для вирішення складних технічних завдань, пов'язаних з розробкою великих проектів та їх складною автоматизацією.

При подачі живлення на контролер Arduino автоматично починається виконання завантаженої в нього програми, а якщо програма відсутня або написана неправильно, виникає збій, який або зупиняє виконання команд, або викликає зависання програми. Номер програми, що виконується, зберігається в спеціальній комірці пам'яті, яка називається лічильником команд. Мова програмування, що використовується в проектах Arduino, заснована на C++. Це одна з найбільш широко використовуваних мов програмування, яка підтримує як низькорівневі команди, так і конструювання складних об'єктів.

На платформі Arduino вивчаються електронні компоненти, їх робота, підключення та програмування. Програмування здійснюється в Arduino IDE на C++ (з деякими модифікаціями). Отримавши базові знання з управління електронними компонентами, слухачі реалізують творчі проекти, складність і функціональність яких обмежені лише фантазією автора.

Arduino дозволяє учням познайомитися зі світом електроніки, зрозуміти принципи роботи електронних компонентів, побачити безліч різноманітних незвичних для них датчиків і пристроїв. Проекти, створені на цій платформі, можна використовувати вдома за призначенням (наприклад, автоматизація побутової техніки), що є додатковою мотивацією до навчання. Заняття з робототехніки на базі Arduino сприяють розвитку політехнічних компетенцій, необхідних у сучасній професійній діяльності в галузі автоматизації та ІТ.

Під час роботи з Arduino користувачі часто хочуть використовувати зовнішні модулі для різних завдань. Це можуть бути датчики для термометра, рідкокристалічний дисплей або підключення до абсолютно іншого комп'ютера чи плати розробника. Зазвичай всі вони однаково підключаються до Arduino, і ця

інструкція представить чотири поширені способи підключення зовнішнього обладнання до вашої плати розробника. Простіше кажучи, «Модуль» у середовищі Arduino – це готова до використання плата з датчиком або пристроєм введення на ньому. Щоб використовувати датчик або пристрій (наприклад, РК-дисплей), вам потрібно кілька додаткових компонентів. Модуль Arduino об'єднує всі необхідні компоненти та з'єднання, необхідні для роботи датчика (або пристрою), і представляє вам готовий до використання продукт.

Перш ніж почати, користувачі повинні з'ясувати, який стандарт зв'язку використовує ваш зовнішній модуль. Найкраще подивитися таблицю або інструкцію пристрою (або подивитися на контакти введення/виводу на самому модулі). Пристрої, які можуть взаємодіяти через послідовне з'єднання, зазвичай мають контакти TX і RX, компоненти, сумісні з I2C, часто відкривають з'єднання SDA і SCL, а пристрої SPI повинні мати контакти з позначками MOSI, MISO і SCK (іноді також CLK або SCLK). Використовується для зв'язку між платою Arduino та комп'ютер або інші пристрої. Усі плати Arduino мають принаймні один послідовний порт (також відомий як UART або USART), а деякі мають кілька.

Наступним кроком після розуміння того, як підключитися до певного пристрою, є знання того, куди підключаються ці контакти на Arduino. Зверніть увагу, що на Arduino Uno, Nano, Mini і Mega контакти 0 і 1 використовуються для зв'язку з комп'ютером. Підключення до цих контактів може призвести до невеликого завантаження. Необхідно звертати увагу на те, що MISO, MOSI і SCK доступні в однаковому положенні майже на всіх Arduino, щоб дозволити дизайн щитів, які працюють на кожній підтримуваній платі. Як згадувалося, послідовний порт – це те, що використовує комп'ютер для зв'язку з Arduino. Він також може надсилати повідомлення на Arduino і виробляти вихідні дані для налагодження. Однак можна використовувати послідовний порт для зв'язку з іншими платами та модулями розробників. Як видно вище, дві плати Arduino підключаються через їхній послідовний порт.

У протоколі зв'язку I2C (рис. 2.1) кожен підключений пристрій повинен мати унікальну адресу, яку зазвичай можна знайти в його таблиці [10].

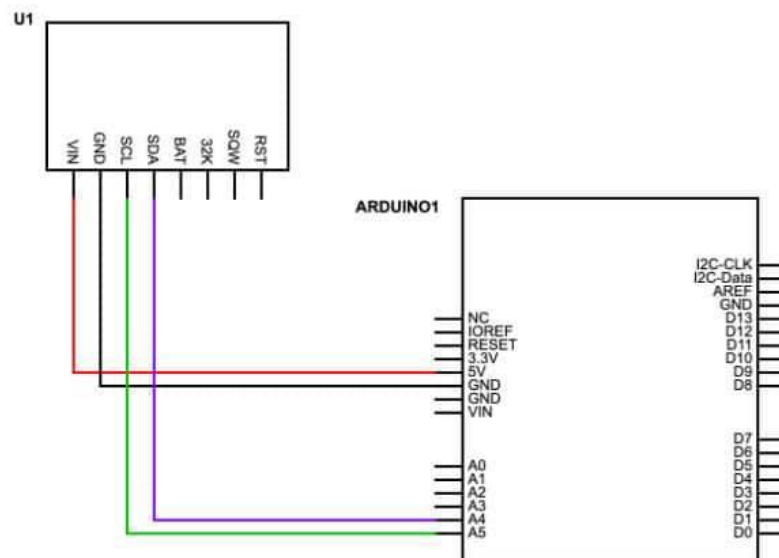


Рисунок 2.2 – Підключення за протоколом I2C [11]

Arduino IDE постачається зі стандартною бібліотекою, яка допомагає в цьому процесі. Зауважте, що в деяких випадках вам знадобляться підтягуючі резистори на лініях SDA і SCL, щоб підтягнути їх високо, коли вони не використовуються. Однак у багатьох модулях вже є відповідні резистори використовуються. Однак у багатьох модулях вже є відповідні резистори.

У SPI (рис. 2.3) завжди є головний пристрій, який керує одним або кількома зовнішніми підлеглими. Як згадувалося вище, зазвичай для цього протоколу потрібні три рядки. Лінія MISO використовується підлеглими для відправки даних на головний пристрій. MOSI призначений для відправки даних до ведених, а сигнал CLK синхронізує всі компоненти. Однак для кожного периферійного пристрою є додаткова лінія SS, що означає підпорядкований вибір. Це дозволяє майстрові вибрати активний периферійний пристрій, щоб дозволити кільком пристроям використовувати одні й ті ж три лінії [12].

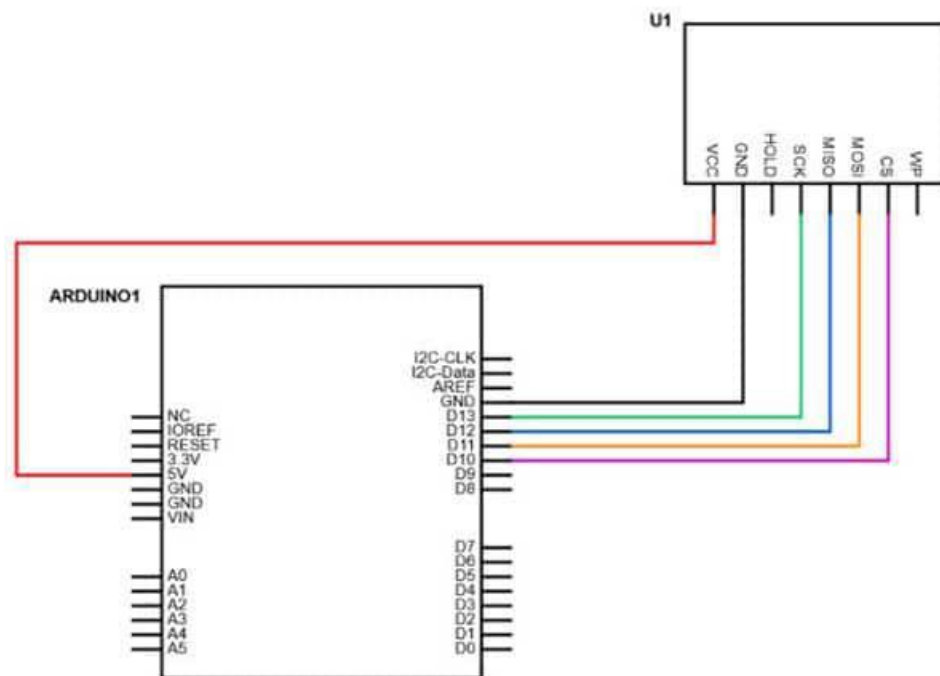


Рисунок 2.3 – Підключення зовнішнього модулю до контактів SPI [12]

Користувачам також доведеться з'ясувати, яка максимальна швидкість передачі їх модуля, чи мають вони надсилати дані в зворотному порядку (спочатку LSB), і як повинен вести себе годинник. Потім вони повинні передати ці значення стандартній бібліотеці Arduino SPI, коли ви хочете спілкуватися з модулем. Як згадувалося на початку, є також четвертий спосіб підключення периферійних пристроїв до вашого Arduino. Деякі пристрої не використовують жоден із методів, які обговорювалися досі. У цих випадках таблиця даних пристрою надасть конкретну інформацію про те, які з'єднання необхідно встановити і як реалізувати сам протокол зв'язку.

## 2.2 Плата драйвера моторів L298P

Найпростіший електродвигун працює тільки на постійному струмі (від батарейки). Струм проходить по рамці, розташованій між полюсами постійного магніту. Взаємодія магнітних полів рамки з струмом і магніту змушує рамку повертатися. Після кожного півоберту колектор перемикає контакти рамки, які підходять до батарейці, і тому рамка обертається.

До мікроконтролера мотори напряму краще не під'єднувати, адже, в основному, мотори працюють зі струмами, напруга яких більша 5В, і в момент виключення або включення мотори створять пікові перепади струму, що може негативно вплинути на плату. Тому для кращого управління моторами використовується такий модуль як драйвер мотора. Драйвер мотора – пристрій, який дозволяє легко та зручно керувати швидкістю та напрямом обертання мотора за допомогою цифрових та ШІМ сигналів [13].

L298P Motor Shield – це плата драйвера двигунів постійного струму, що використовує мікросхему потужного драйвера електродвигунів L298P, яка може безпосередньо управляти двома двигунами постійного струму; струм через навантаження – до 2 ампер. Вихідні інтерфейси управління двигунами використовують вісім високошвидкісних діодів в якості захисту. Дана плата може бути встановлена безпосередньо на плату Arduino (рис. 2.4).

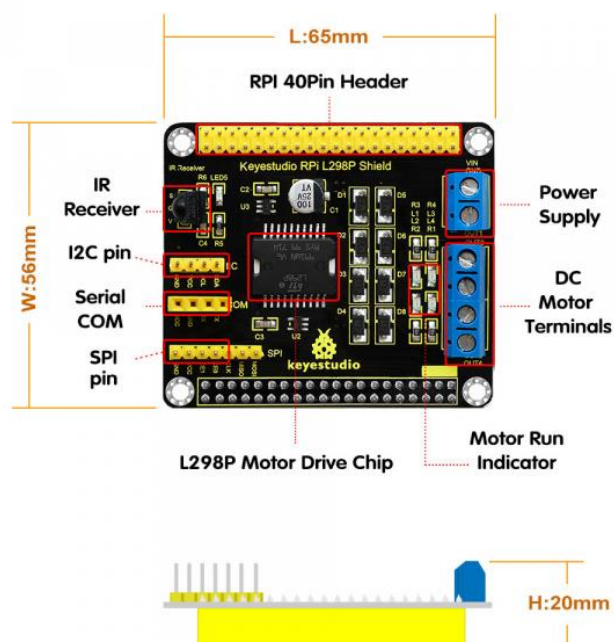


Рисунок 2.4 – L298P Motor Shield [13]

Основні характеристики L298P Motor Shield:

- Вхідна напруга логічної частини: 5В;
- Вхідна напруга рушійної частини: 4,8 ~ 24 В;

- Робочий струм логічної частини  $I_{ss} \leq 36\text{mA}$ ;
- Робочий струм провідної частини  $I_o \leq 2\text{A}$ ;
- Максимальна розсіювана потужність:  $25\text{Вт}$  ( $T=75\text{ }^\circ\text{C}$ );
- Робоча температура:  $-25\text{ }^\circ\text{C} \sim +130\text{ }^\circ\text{C}$ .

Мотор керується трьома портами – двома цифровими та одним ШІМ. Мотор може обертатися за годинниковою стрілкою або проти неї. За це відповідають цифрові порти, які на драйвері мотора підключаються до контактів IN (рис. 2.5). За швидкість відповідає ШІМ порт і він підключається до EN контакту і може задавати діапазон значень 0-255.

IN1 - **високий**, IN2 - **низький** = обертання в одну сторону;  
 IN1 - **низький**, IN2 - **високий** = обертання в іншу сторону;  
 IN1 - **низький**, IN2 - **низький** = обертання немає;  
 IN1 - **високий**, IN2 - **високий** = обертання немає.

Рисунок 2.5 – Комбінації сигналів для мотора [13]

На рисунку 2.6 показано принципову схему плати L298P

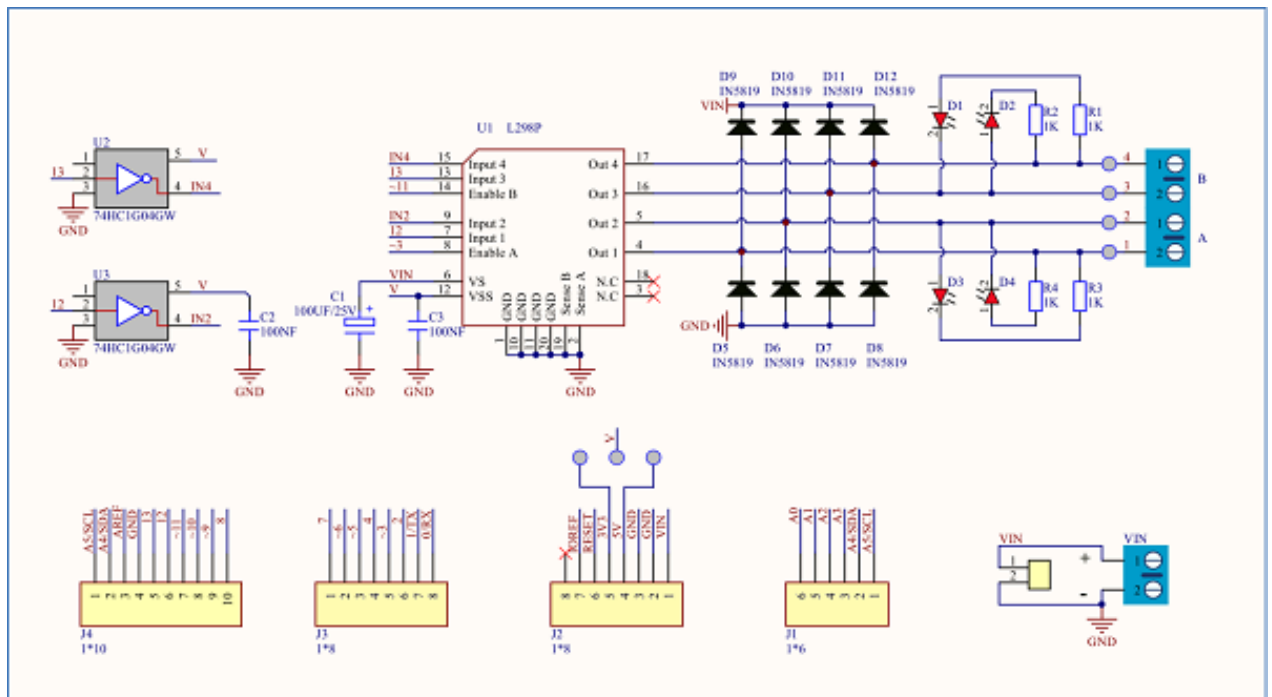


Рисунок 2.6 – Принципова схема плати L298P

### 2.3 Модуль GY-521

Модуль GY-521 (рис. 2.7) на мікросхемі MPU-6050 (рис. 2.8) – це триосний гіроскоп і акселерометр, поширений на три координати, а також датчик температури навколишнього середовища.

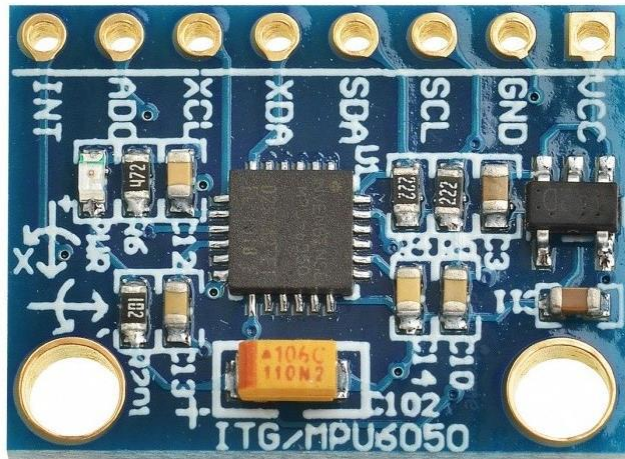


Рисунок 2.7 – Зовнішній вигляд модуля GY-521 [14]

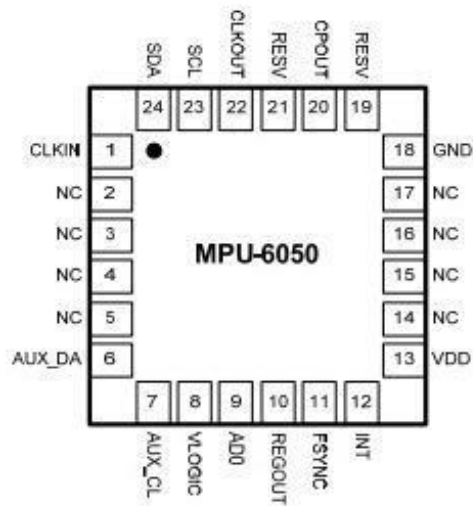


Рисунок 2.8 – Мікросхема MPU-6050 [14]

Крім цього на платі самого модуля GY-521 розміщені необхідні для надійного функціонування підтягуючі резистори. Обмін даними з контролером управління здійснюється за допомогою шини I2C. Необхідна напруга живлення для модуля і всіх наявних на ньому датчиків 3.3V.

Для найбільш точного трекінгу швидких або повільних рухів, можливо, самостійно запрограмувати необхідні режими:  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ ,  $\pm 2000$  гр/сек для гіроскопа, а також  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$  для акселерометра [14].

Специфікація і характеристики модуля GY-521:

Мікросхема: MPU-6050 (гіроскоп 3-осьовий + акселерометр на 3 координати);

- Напруга живлення модуля: від 3.3V до 5V (DC).
- Чіп MPU-6050: 16 бітний АЦП, 16 біт вивід даних.
- Гіроскоп діапазон:  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ ,  $\pm 2000$  гр / с.
- Акселерометр діапазон:  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$ .
- Зв'язок з контролером за стандартним комунікаційним протоколом I2C.
- Розміри модуля: 15x20 мм. .
- Вага модуля: 5 грам. [15]

## 2.4 Модуль NRF24L01

Модуль NRF24L01 використовує діапазон 2,4 ГГц і може працювати зі швидкістю передачі від 250 кбіт/с до 2 Мбіт/с. Якщо використовується у відкритому космосі та з меншою швидкістю передачі, його дальність може сягати 100 метрів, сам модуль зображено на рисунку 2.10.

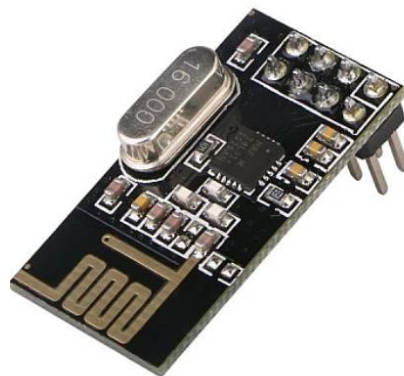


Рисунок 2.10 – Модуль NRF24L01 [16]

Модуль може використовувати 125 різних каналів (рис. 2.11), що дає можливість мати в одному місці мережу з 125 незалежно працюючих модемів. Кожен канал може мати до 6 адрес, або кожен блок може спілкуватися одночасно з до 6 іншими блоками.

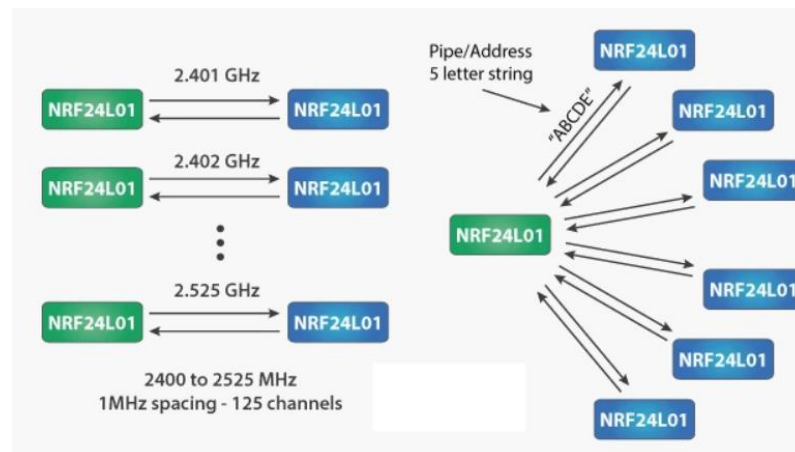


Рисунок 2.11 – Модуль NRF24L01 [16]

Енергоспоживання цього модуля становить приблизно 12 мА під час передачі, що навіть менше, ніж у одного світлодіода. Робоча напруга модуля становить від 1,9 до 3,6 В (рис. 2.12), але добре, що інші висновки переносять логіку 5 В, тому ми можемо легко підключити його до Arduino без використання перетворювачів логічного рівня [17].



Рисунок 2.12 – Робоча напруга модуля [16]

Три з цих контактів призначені для зв'язку SPI, і їх потрібно підключити до контактів SPI Arduino, але зауважте, що кожна плата Arduino має різні контакти SPI. Виводи CSN та CE можуть бути підключені до будь-якого цифрового виводу плати Arduino, і вони використовуються для переведення модуля в режим очікування або активного режиму, а також для перемикання між режимом передачі або команди. Останній штифт – це штифт переривання, який не потрібно використовувати.

## РОЗДІЛ 3

### РОЗРОБКА СИСТЕМИ КЕРУВАННЯ МОБІЛЬНИМ РОБОТОМ

#### 3.1 Середовище Arduino IDE

IDE (або інтегроване середовище розробки) – це програма, призначена для розробки програмного забезпечення. Як випливає з назви, IDE об'єднує кілька інструментів, спеціально призначених для розробки. Ці інструменти зазвичай включають редактор, призначений для роботи з кодом (наприклад, підсвічування синтаксису і автодоповнення); інструменти збірки, виконання та налагодження; і певну форму системи управління версіями.

Більшість IDE підтримують безліч мов програмування і мають багато функцій, через що можуть бути великими, займати багато часу для завантаження і установки і вимагають глибоких знань для правильного використання. З іншого боку, є редактори коду, які представляють собою текстовий редактор з підсвічуванням синтаксису і можливостями форматування коду. Більшість хороших редакторів коду можуть виконувати код і використовувати відладчик, а кращі навіть можуть взаємодіяти з системами управління версіями. У порівнянні з IDE, хороший редактор коду, як правило, легковагі і швидше, але часто ціною меншою функціональності. Набір функцій різних середовищ може відрізнитися, але є набір базових речей, що спрощують програмування:

- збереження файлів. Якщо IDE або інший редактор не дають можливості зберегти роботу і пізніше все відкрити в тому ж стані, в якому воно було під час закриття, то не дасть змогу працювати;
- запуск коду з середовища. Те ж саме, якщо потрібно вийти з середовища для запуску коду, то це не більше, ніж простий текстовий редактор;
- підтримка налагодження. Можливість крок за кроком виконати код є базовою функцією всіх IDE і більшості хороших редакторів коду; підсвічування синтаксису. Можливість швидко знайти ключові слова, змінні та інше робить читання і розуміння коду на порядок простіше;
- автоматичне форматування коду. Будь-який редактор або IDE, який

дійсно таким є, розпізнає двокрапку після `while` або `or` вираження і автоматично зробить відступ на наступному рядку.

Зрозуміло, є безліч інших функцій, від які були б тільки на користь, але наведені вище - основні функції, якими повинна володіти комфортне середовище розробки.

Arduino – це інструмент для тестування електронних пристроїв, який більш щільно працює з навколишнього фізичного середовища, ніж стандартні комп'ютери. Це платформа, призначена для керування фізичними процесами з використанням ЕВМ з відкритим програмним кодом, побудована на простій печатній платі с сучасним середовищем для написання програмного забезпечення. Плати Arduino будуються на основі мікроконтролерів фірми Atmel, а також елементів обв'язки для програмування інтеграції з іншими схемами. На платах присутній лінійний стабілізатор напруги +5 В або +3,3 В. Тактування здійснюється на частотах 8, 16 або 87 МГц кв. У мікроконтролер попередньо прошивається завантажувач, тому зовнішній програматор не потрібний. На концептуальному рівні всі плати програмуються через RS-232. Інтегроване середовище розробки

Arduino – це кросплатформовий застосунок на Java, що включає в себе редактор кода, компілятор і модель передачі прошивки на плату. Середовище розробки засноване на мові програмування Processing і спроектована для програмування новачками, близько не знайомими з розробкою програм, це - мова C++, ще доповнена деякими бібліотеками.

Найпершою, а, найчастіше, і єдиною програмою для початківців працювати з контролером Ардуїно стає IDE Arduino – інтегроване середовище розробки від творців платформи, застосовується на Windows, MacOS та Linux. Це середовище розроблено на Cі та C++. Призначена вона для створення та завантаження програм на Arduino-сумісні плати, а також на плати інших виробників, для розробки прошивок та завантаження їх у пам'ять мікроконтролера.

Середовище розробки Arduino IDE (рис. 3.1) складається із набору специфічних бібліотек та драйверів, необхідних для коректної роботи з контролером. За допомогою вбудованих бібліотек завжди можна підключити додатковий датчик або дисплей до контролера. Можна перевірити код на наявність помилок. Програма вкаже точне місце помилки в коді або де було введено неправильні дані. Якщо їх не виправити, програма просто не зможе скомпілювати код, і не вдасться залити скетч в контролер.

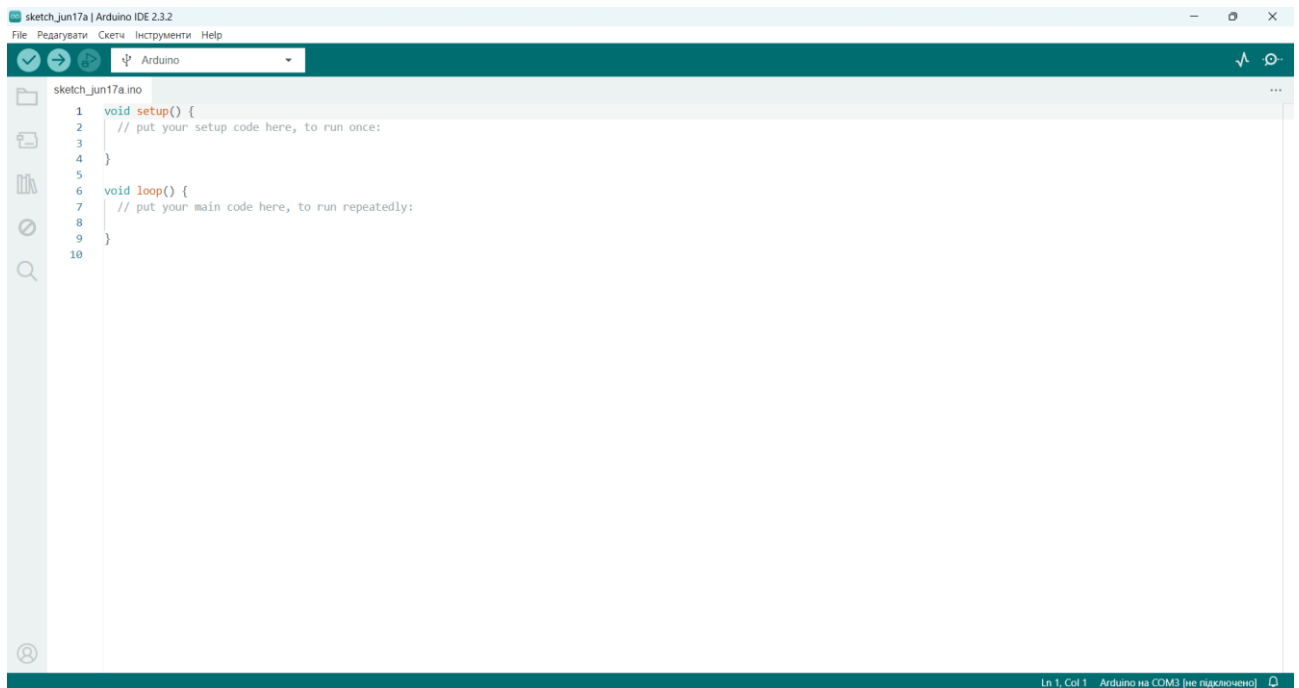


Рисунок 3.1 – IDE Arduino

## 3.2 Апаратна розробка

Щоб зрозуміти, як працює машина керування жестами Arduino, давайте розділимо цей проект на дві частини. Перша частина – це частина передавача (дистанційна), в якій датчик акселерометра MPU6050 безперервно надсилає сигнали до приймача (робота) через Arduino та передавач nRF.

Друга частина – це приймальна частина (автомобіль-робот), у якій nRF-приймач отримує передані дані та надсилає їх до Arduino, який далі обробляє їх і відповідно переміщує робота.

На рисунку 3.2 показана структурна схема проекту.

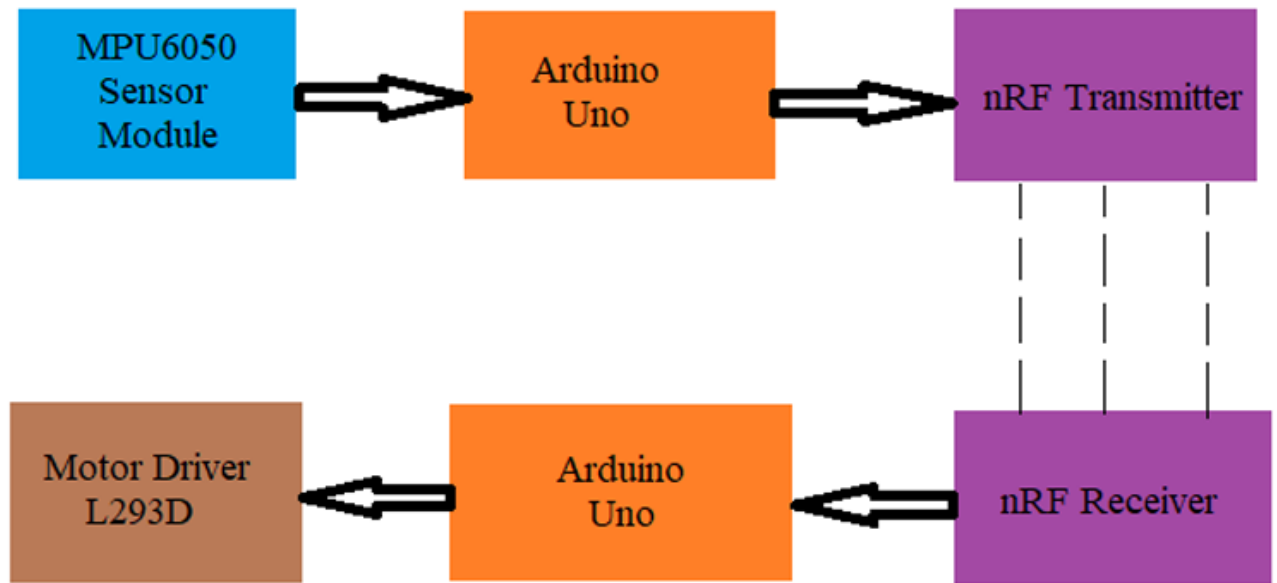


Рисунок 3.2 – Структурна схема проекту

Датчик акселерометра MPU6050 зчитує координати XYZ і надсилає координати на Arduino. Для цього проекту нам потрібні лише координати X і Y. Потім Arduino перевіряє значення координат і надсилає дані на передавач nRF. Передані дані приймаються приймачем nRF. Приймач надсилає дані до Arduino приймача. Arduino передає дані в мікросхему драйвера двигуна, і драйвер двигуна повертає двигуни в потрібному напрямку.

Цей робот, керований жестами рук, використовує апаратне забезпечення Arduino, розділений на дві частини:

1. Передавач.
2. Приймач.

Передавач проекту складається з акселерометра та гіроскопа MPU6050, трансивера nRF24L01 і Arduino Uno. Arduino постійно отримує дані від MPU6050 і надсилає їх на nRF-передавач. Радіочастотний передавач передає дані в навколишнє середовище.

Принципова схема для секції передавача наведена нижче (рис. 3.3).

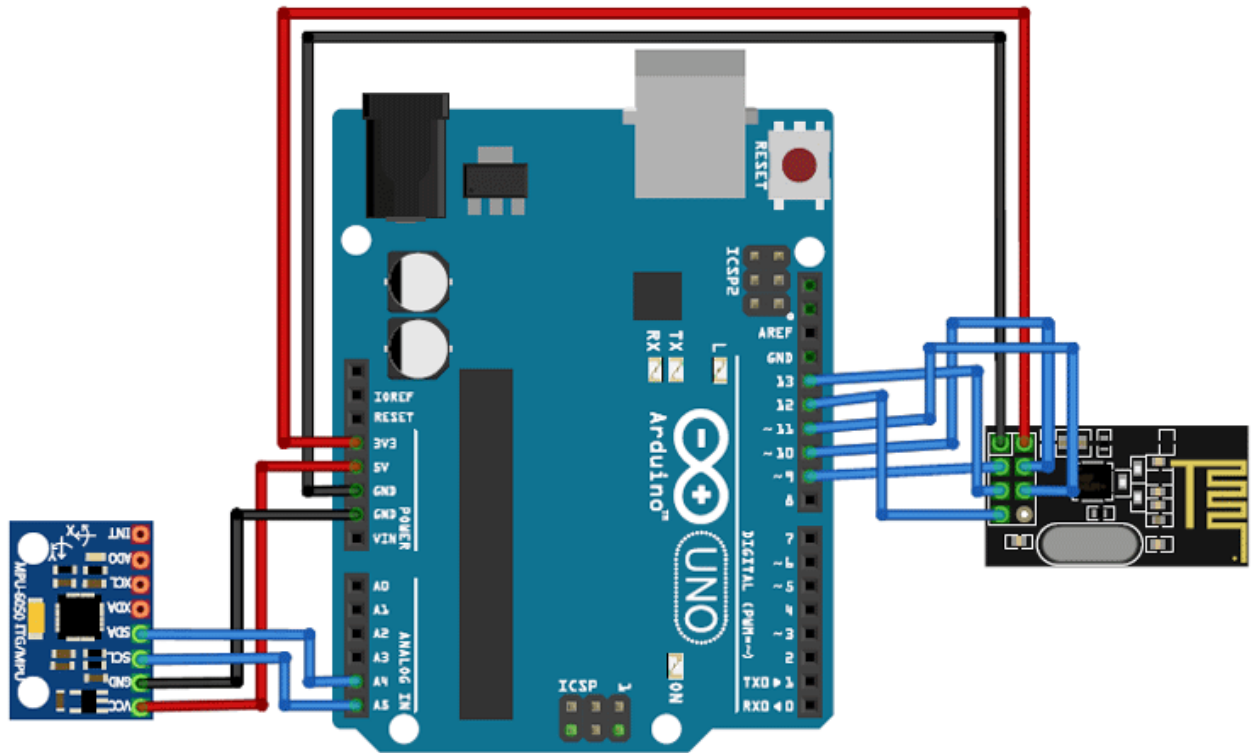


Рисунок 3.3 – Принципова схема передавача

Підключення плати Arduino Uno до датчика акселерометра MPU6050 та радіоприймача RF24L01 (табл. 3.1).

Таблиці 3.1 – Підключення датчиків до плати

nRF24L01	Arduino Uno
1	2
VCC	3,3 B
GND	GND
CE	Pin 9
CSN	Pin10
SCK	Pin 13
MOSI	11
MISO	12
MPU6050	Arduino Uno
VCC	5B

## Продовження таблиці 3.1

1	2
GND	GND
SCL	Булавка A5
ПДР	Булавка A4
ІНТ	Pin 2

Передавач цього проекту складається з акселерометра та гіроскопа MPU6050, трансивера nRF24L01 і Arduino Uno. Arduino постійно отримує дані від MPU6050 і надсилає їх на nRF-передавач. Радіочастотний передавач передає дані в навколишнє середовище (рис. 3.4).

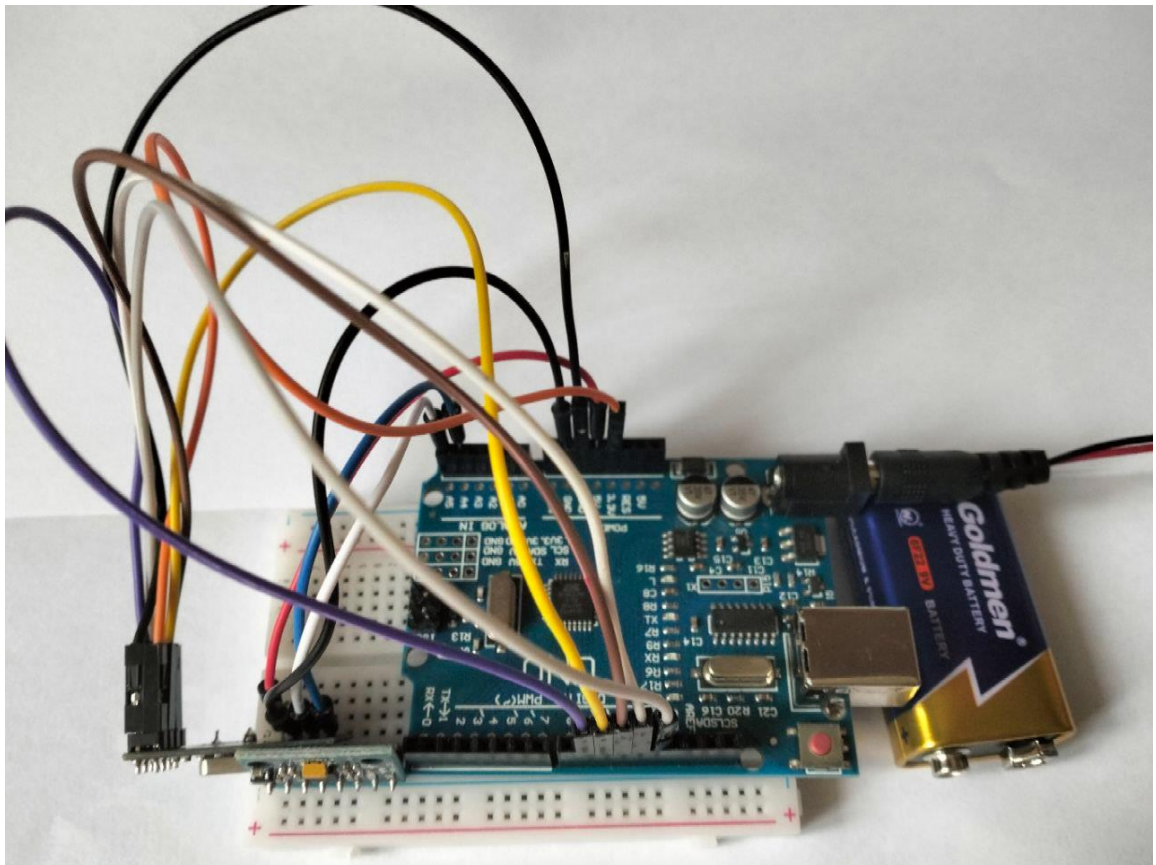


Рисунок 3.4 – Передавач

На рисунку 3.5 наведена ринципова схема секції приймача.

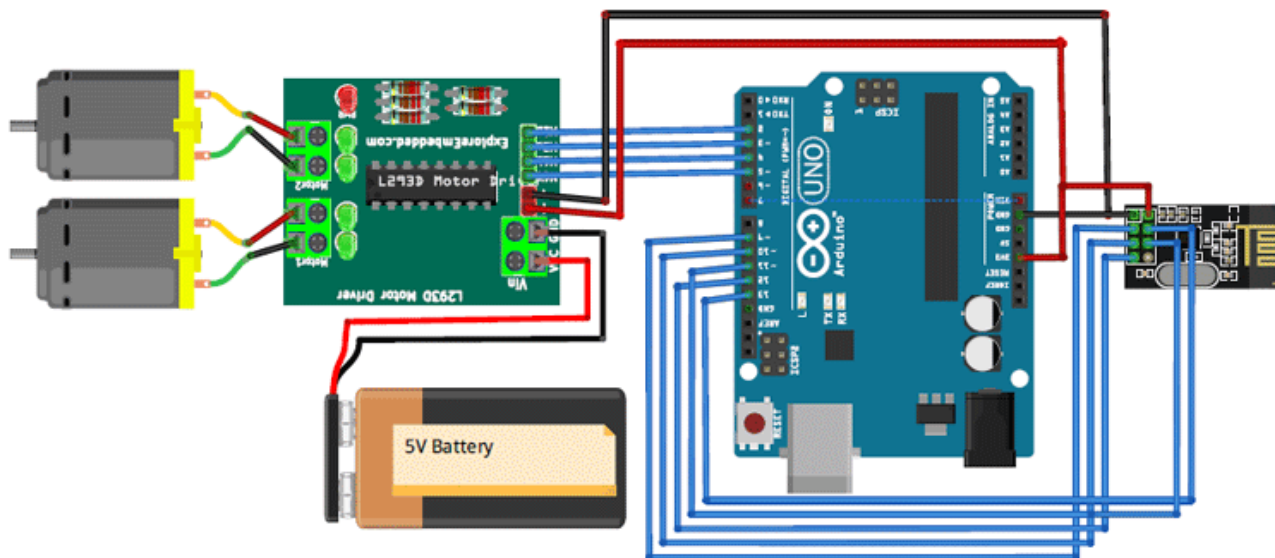


Рисунок 3.5 – Принципова схема приймача

Підключення плати Arduino Uno до датчика акселерометра MPU6050 та радіоприймача RF24L01 (табл. 3.2).

Таблиці 3.2 – Підключення датчиків до плати

RF24L01	Arduino Uno
1	2
VCC	3,3 B
GND	GND
ЦЕ	Pin 9
CSN	Pin10
SCK	Pin 13
ДИМ	11
MISO	12
Двигун	Arduino Uno
VCC	5B

## Продовження таблиці 3.2

1	2
GND	GND
ІНП А1	2
ІНП А2	3
ІНП В1	4
ІНП В2	5

Приймач цього робота, керованого жестами, складається з Arduino Uno, трансивера nRF24L01, 2 двигунів постійного струму та модуля драйвера двигуна. Приймач NRF24L01 отримує дані від передавача та надсилає їх на Arduino. Потім відповідно до отриманих сигналів Arduino рухає двигуни постійного струму (рис. 3.5).

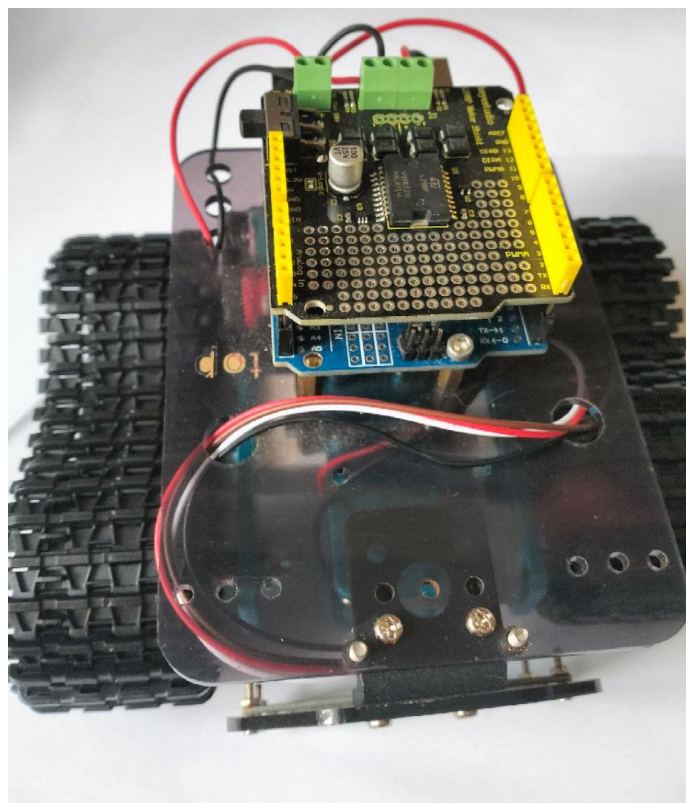


Рисунок 3.5 – Приймач

### 3.3 Програмна розробка

У програмі передавача Arduino зчитує дані з MPU6050 і надсилає їх на передавач nRF 24L01. Спочатку додамо необхідні файли бібліотеки, а саме:

- SPI.h.
- nRF24L01.h.
- Wire.h.
- MPU6050.h.

Потім визначимо змінні для даних гіроскопа та акселерометра MPU6050. Тут будуть використані лише дані акселерометра (лістинг 3.1).

#### Лістинг 3.1 – Задання змінних

---

```
MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
}
```

---

Кінець лістингу 3.1

Визначимо адреси радіоканалу для контактів CN і CSN передавачів зв'язку та nRF (лістинг 3.2).

#### Лістинг 3.2 – Визначення адреси радіоканалу

---

```
const uint64_t pipeOut = 0xE8E8F0F0E1LL;
радіо RF24 (9, 10);
```

---

Кінець лістингу 3.2

Всередині функції void setup() запустить послідовний монітор. А також ініціалізувати дротовий і радіозв'язок. Radio.setDataRate використовується для встановлення швидкості передачі даних (лістинг 3.3).

### Лістинг 3.3 – Функція void setup()

---

```
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
  radio.begin();
  radio.setAutoAck(false);
  radio.setDataRate(RF24_250KBPS);
  radio.openWritingPipe(pipeOut);
}
```

---

Кінець лістингу 3.3

Зчитуємо дані датчика MPU6050. Тут ми використовуємо лише дані акселерометра в напрямку X і Y (лістинг 3.4).

### Лістинг 3.4 – Зчитуємо дані датчика MPU6050

---

```
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
data.X = map(ax, -17000, 17000, 0, 255);
data.Y = map(ay, -17000, 17000, 0, 255);
```

---

Кінець лістингу 3.4

Передамо дані датчика за допомогою функції *radio.write* (лістинг 3.5).

### Лістинг 3.5 – Функції radio.write

---

```
radio.write(&data, sizeof(MyData));
```

---

Кінець лістингу 3.5

Додами необхідні бібліотеки до програми приймача (лістинг 3.6).

### Лістинг 3.6 – Додаємо бібліотеки

---

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
```

---

Кінець лістингу 3.6

Визначаємо адреси радіоканалу для контактів CN і CSN передавачів зв'язку та nRF (лістинг 3.7).

#### Лістинг 3.7 – Визначаємо адреси радіоканалу

---

```
const uint64_t pipeln = 0xE8E8F0F0E1LL;
радіо RF24 (9, 10);
```

---

Кінець лістингу 3.7

Визначимо лівий і правий контакти двигуна постійного струму (лістинг 3.8).

#### Лістинг 3.8 – Визначаємо лівий і правий контакти двигуна

---

```
const int IN1 = 2;
const int IN2 = 3;
const int IN3 = 4;
const int IN4 = 5;
```

---

Кінець лістингу 3.8

Тепер перевіримо, чи доступне радіо чи ні. Якщо так, то прочитайте дані (лістинг 3.9).

#### Лістинг 3.9 Визначаємо чи доступне радіо

---

```
f ( radio.available() ) {
    radio.read(&data, sizeof(MyData));
```

---

Кінець лістингу 3.9

Тепер порівняємо отримані дані і заведіть двигуни відповідно до умов (лістинг 3.10).

### Лістинг 3.10 – Порівняння даних

---

```
if (data.Y < 80) { //Назад
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}
```

---

Кінець лістингу 3.10

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проаналізовано стан сучасної робототехніки. та проведений аналіз сфер застосування мобільних роботів. Застосування робототехніки в різних сферах, пов'язаних з вирішенням спеціальних завдань, вимагає максимальної спрощення способів взаємодії людини і робота.

В другому розділі зроблено аналіз та вибір апаратної платформи. Були сформовані мінімальні вимоги щодо апаратної частини проекту. Дослідження апаратних систем типу Single-board computer дало розуміння можливостей цих міні комп'ютерів. Так як система не потребує великих математичних обчислень, можливості апаратних систем типу Single-board computer задовольняють мінімальні вимоги. Так як на ринку одноплатних комп'ютерів представлено достатньо багато продуктів був проведений аналіз цього ринку і більш детально зупинився на середовищі Arduino.

В третьому розділі наведений опис програмної частини системи, вибір середовища розробки та основні елементи. Середовище Arduino дозволяє реалізувати всі необхідні завдання які поставлено перед системою для розробника максимально просто і швидко.

Напрямами подальшого удосконалення розробки можуть бути: додавання заздалегідь запрограмованих інтерфейсів керування для існуючих роботів, або робототехнічних систем; додавання можливості керування пристроями за допомогою мережевого зв'язку через мережу Internet; розробка більш гнучкого інтерфейсу, для надання можливості користувачу підлаштовувати його під свої цілі; розробка більш досконалої системи команд; додавання програмного протоколу передачі даних, для унеможливлення втрати пакетів та більшої безпеки під час передачі даних.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Будова та система керування мобільної платформи на базі робота павука. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2018/05/148-1.pdf> (дата звернення: 28.02.2024).
2. Основи робототехніки. URL: [https://ec.europa.eu/programmes/erasmus-plus/project-result-content/f49ee634-1909-4c5d-ab780ff34a693f94/book\\_Robotics.pdf](https://ec.europa.eu/programmes/erasmus-plus/project-result-content/f49ee634-1909-4c5d-ab780ff34a693f94/book_Robotics.pdf) (дата звернення: 28.02.2024).
3. Mobile robotics (Robot). Module 1. URL: <https://learn.open2study.com/mod/youtube/view.php?id=79916> (дата звернення: 28.02.2024).
4. Siegward, R Introduction to Autonomous Mobile Robots. URL: [https://www.ucg.ac.me/skladiste/blog\\_13268/objava\\_56689/fajlovi/Introduction%20o%20Autonomous%20Mobile%20Robots%20book.pdf](https://www.ucg.ac.me/skladiste/blog_13268/objava_56689/fajlovi/Introduction%20o%20Autonomous%20Mobile%20Robots%20book.pdf) (дата звернення: 28.02.2024).
5. Економічна правда. URL: <https://epravda.com.ua/tags/ritejl/> (дата звернення: 28.02.2024).
6. HUAWEI. URL: <https://4huawei.ru/about-huawei> (дата звернення: 15.03.2024).
7. Sebastian Thrun: Google's driverless car. URL: [http://www.ted.com/talks/sebastian\\_thrun\\_google\\_s\\_driverless\\_car](http://www.ted.com/talks/sebastian_thrun_google_s_driverless_car) (дата звернення: 20.03.2024).
8. Elfes A. Sonar-based real-world mapping and navigation. URL: <https://ieeexplore.ieee.org/abstract/document/1087096/> (дата звернення: 20.03.2024).
9. Робототехніка та мехатроніка. URL: [https://pdf.lib.vntu.edu.ua/books/2021/Tsvirkun\\_224.pdf](https://pdf.lib.vntu.edu.ua/books/2021/Tsvirkun_224.pdf) (дата звернення: 10.04.2024).
10. Arduino Uno. URL: [https://uk.wikipedia.org/wiki/Arduino\\_Uno](https://uk.wikipedia.org/wiki/Arduino_Uno) (дата звернення: 18.04.2024).

11. How To Connect a Thermal Printer to an Arduino Board.  
<https://www.digikey.com.br/en/maker/tutorials/2022/how-to-connect-a-thermal-printer-to-an-arduino-board> (дата звернення: 18.04.2024).

12. Короткий вступ до інтерфейсу SPI (Serial Peripheral Interface). URL:  
<https://doc.arduino.ua/prog/SPI> (дата звернення: 20.04.2024).

13. Драйвер двигунів L298N URL: <https://3d-diy.ru/wiki/arduino-moduli/drayver-dvigatelya-l298n/#Techno> (дата звернення: 25.04.2024).

14. Гіроскоп MPU-6050, акселерометр. URL:  
[http://carduino.ru/product\\_info.php?products\\_id=1954/](http://carduino.ru/product_info.php?products_id=1954/) (дата звернення: 25.04.2024).

15. Модуль 3-х осьового гіроскопа і акселерометра GY-521 MPU-6050.-  
URL: <http://arduino-kit.ru/> (дата звернення: 25.04.2024).

16. Everything about nRF24L01 transceiver. URL:  
<https://www.flyrobo.in/blog/nrf24l01-transceiver> (дата звернення: 20.05.2024).

# ДОДАТКИ

## Додаток А

### Код робота, керованого жестами за допомогою Arduino

#### Побічна програма передавача

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include «Wire.h»
#include «I2Cdev.h»
#include «MPU6050.h»

//Визначення змінних для даних гіроскопа та акселерометра
MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;

const uint64_t pipeOut = 0xE8E8F0F0E1LL;

радіо RF24 (9, 10); // Виводи CN і CSN nrf

struct MyData {
  byte X;
  байт Y;
};

дані MyData;

void setup()
{
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();

  radio.begin();
  radio.setAutoAck(false);
  radio.setDataRate(RF24_250KBPS);
  radio.openWritingPipe(pipeOut);
}

void loop()
{
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  data.X = map(ax, -17000, 17000, 0, 255); //Надіслати дані даних осі X.Y
  = map(ay, -17000, 17000, 0, 255); //Надіслати дані осі Y

  затримка (50);
  radio.write(&data, sizeof(MyData));

  Serial.print(«Вісь X = «);
  Serial.print(дані.X);
  Serial.print(« «);
  Serial.print(«Вісь X = «);
  Serial.println(дані.X);
}

```

#### Програма сторони приймача

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
const uint64_t pipeIn = 0xE8E8F0F0E1LL;

радіо RF24 (9, 10);

```

```

// Лівий і правий контакти двигуна
const int IN1 = 2;
const int IN2 = 3;
const int IN3 = 4;
const int IN4 = 5;

struct MyData {
  byte X;
  байт Y;
};

дані MyData;

void setup()
{
  Serial.begin(9600);
  radio.begin();
  radio.setAutoAck(false);
  radio.setDataRate(RF24_250KBPS);
  radio.openReadingPipe(1, pipeIn);
  radio.startListening();
}

void recvData()
{
  if ( radio.available() ) {
    radio.read(&data, sizeof(MyData));

    if (data.Y < 80) { //Зворотний
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    }

    if (data.Y > 145) { //forward
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, LOW);
    }

    if (data.X > 155) { //правий поворот
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    }

    if (data.X < 80) { //ліворуч
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, LOW);
    }

    if (data.X > 100 && data.X < 170 && data.Y > 80 && data.Y < 130) { //зупинка автомобіля
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, LOW);
    }
  }
}

```

```
void loop()
{
  recvData();
  Serial.print(«X: «);
  Serial.print(дані.X);
  Serial.print(« «);
  Serial.print(«Y: «);
  Serial.print(data.Y);
  Serial.print(«\n»);
}
```