

ЗМіністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
Розробка реконфігурованого мобільного робота для навчання студентів
(тема)

Виконав: здобувач 2 року навчання,
групи КТРСм-23-2
Дмитрієв Д.В.
(прізвище, ініціали)

Спеціальність 174 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та
робототехнічні системи
(повна назва освітньої програми)

Керівник проф. каф. КІТАР Євсєєв В.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Невлюдов І.Ш.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 174 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та робототехнічні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Дмитрієву Д.В.
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка реконфігурованого мобільного робота для
навчання студентів

затверджена наказом університету від 25.11.2024 р. № 1239

2. Термін подання здобувачем роботи до екзаменаційної комісії 29. 01.2025р.

3. Вихідні дані до роботи

3.1 Дистанційна система керування

3.2 Програмне середовище для розробки ArduinoIDE

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Вступ

4.2 Аналіз конструкцій реконфігурованих мобільних роботів

4.3 Розробка структури та системи керування реконфігурованим мобільним
роботом

4.4 Розробка програмного забезпечення

4.5 Експериментальне дослідження та аналіз отриманих результатів

4.6 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

Демонстраційний матеріал, представлений у форматі презентації
PowerPoint(*.ppt) Формату А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання, вступ	21.10.24 – 25.10.24	виконано
2	Аналіз сучасних прототипів реконфігурованих мобільних роботів	25.10.24 – 29.10.24	виконано
3	Вибір апаратних модулів	04.11.24 – 08.11.24	виконано
4	Розробка макету реконфігурованого мобільного робота	15.11.24 – 02.12.24	виконано
5	Розробка схеми підключення	03.12.24 – 10.12.24	виконано
6	Розробка загального алгоритму роботи	12.12.24 – 19.12.24	виконано
7	Оформлення пояснювальної записки	20.12.2024 – 27.12.24	виконано
8	Подання роботи до ЕК	29.01.25	виконано

Дата видачі завдання 11 листопада 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф.каф.КІТАР Євссєв В.В.
(підпис) (посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«30» грудня 2024 р.

A handwritten signature in blue ink, appearing to be 'D. V. Dmitriyev', written in a cursive style.

Дмитрієв Д.В.

РЕФЕРАТ

Пояснювальна записка: 103 с., 1 табл., 65 рис., 3 дод., 26 джерел.

"ЗРОБИ САМ", ОСВІТА, ВІДКРИТЕ ДЖЕРЕЛО, АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ, ПРОЕКТНЕ НАВЧАННЯ, ШВИДКЕ ПРОТОТИПУВАННЯ, РОБОТОТЕХНІКА.

Об'єкт дослідження – процес керування реконфігурованим мобільним роботом.

Предмет дослідження – методи, конструкції, алгоритми та програмні забезпечення системи керування реконфігурованим мобільним роботом.

Мета роботи – підвищення ефективності проведення освітнього процесу за рахунок уніфікації модулів реконфігурованого мобільного робота.

В ході написання кваліфікаційної роботи було проведено аналіз існуючих реконфігурованих мобільних роботів, їх конструкції і системи управління. На основі отриманих даних було розроблено структурну схему, проведено аналіз та вибір апаратних модулів і зібрано макет реконфігурованого мобільного робота. Розроблено загальний алгоритм роботи реконфігурованого мобільного робота та алгоритм роботи керуючих пристроїв. Були передбачені заходи з охорони праці в приміщенні лабораторії, у якій планується використовувати розроблений робот. Написано програму для дистанційного керування зібраним роботом і проведено експеримент.

Розроблений макет може бути використаний в закладах освіти для вивчення основ робототехніки та програмування з покроковим виконанням завдань.

Результати кваліфікаційної роботи можна віднести до Цілі сталого розвитку 4 "Якісна освіта", п. 4.4 "Підвищити якість вищої освіти та забезпечити її тісний зв'язок з наукою, сприяти формуванню в країні міст освіти та науки".

ABSTRACT

Explanatory note: 103 p., 1 table, 65 figures, 3 appendix, 26 sources.

"DIY", EDUCATION, OPEN SOURCE, HARDWARE, PROJECT-BASED LEARNING, RAPID PROTOTYPING, ROBOTICS.

The object of the study is the control system of a reconfigured mobile robot.

The subject of the study is models and methods of controlling a reconfigured mobile robot based on wireless control.

The purpose of the qualification work is to develop a laboratory model and control system for a reconfigured mobile robot.

During the writing of the qualification work, an analysis of existing reconfigured mobile robots, their designs and control systems was carried out. Based on the data obtained, a structural diagram was developed, an analysis and selection of hardware modules were carried out, and a model of a reconfigured mobile robot was assembled. A general algorithm for the operation of the reconfigured mobile robot and an algorithm for the operation of control devices were developed. Occupational safety measures were provided in the laboratory premises in which the developed robot is planned to be used. A program was written for remote control of the assembled robot and an experiment was conducted.

The developed model can be used in educational institutions to study the basics of robotics and programming with step-by-step task completion.

The results of the qualification work can be attributed to Sustainable Development Goal 4 "Quality Education" item 4.4 "Improve the quality of higher education and ensure its close connection with science, promote the formation of cities of education and science in the country."

ЗМІСТ

Перелік скорочень.....	9
Вступ	10
1 Аналіз конструкцій реконфігурованих мобільних роботів	12
1.1 Аналіз конструкцій реконфігурованих мобільних роботів.	12
1.2 Аналіз аналогічних рішень	17
1.3 Аналіз методів керування реконфігурованим мобільним роботом.....	25
1.4 Висновки до першого розділу	29
2 Розробка структури та системи керування реконфігурованим мобільним роботом	30
2.1 Аналіз апаратних модулів	30
2.2 Структура модульного реконфігурованого робота.....	33
2.3 Розробка загальної структурної схеми	47
2.4 Можливі схеми підключення.....	49
2.5 Складання рівняння двигуна постійного струму з незалежним збудженням	51
2.6 Складання макету мобільного реконфігурованого робота.....	56
2.7 Висновки до другого розділу	59
3 Розробка програмного забезпечення	61
3.1 Вибір середовища розробки.....	61
3.2 Розробка загального алгоритму	65
3.3 Приклад програмної реалізації функцій String та часу	76
3.4 Висновки до третього розділу	81
4 Експериментальне дослідження та аналіз отриманих результатів.....	83
4.1 Експериментальне дослідження рухом мобільного робота	83
4.2 Аналіз експериментального дослідження	90
4.3 Розрахунок штучного освітлення робочого місця монтажу та експлуатації реконфігурованим мобільним роботом.....	95

4.4 Висновки до четвертого розділу.....	98
Висновки.....	100
Перелік джерел посилання.....	102
Додаток А Скетч програми для Arduino.....	105
Додаток Б Публікація. Розробка реконфігурованого мобільного робота.....	105
Додаток В Демонстраційний матеріал у вигляді презентації	119

ПЕРЕЛІК СКОРОЧЕНЬ

- АЦП – аналого-цифровий перетворювач;
ЗП – захоплювальний пристрій;
ПР – промисловий робот;
ПК – персональний комп'ютер;
САУ – система автоматичного управління;
САР – система автоматичного регулювання;
ТЗ – технічне завдання.

ВСТУП

Сучасне виробництво та індустріальні процеси відзначаються постійними змінами та швидким технологічним розвитком. Для ефективного використання робочих ресурсів та підвищення продуктивності, підприємства повинні шукати інноваційні рішення, зокрема, в області автоматизації. Однак, існує проблема несталості вимог до роботизованих систем. Несталість вимог може виникати через різноманітні фактори, такі як, зміни у виробничому процесі, введення нових завдань чи технологічні інновації. Існує потреба у гнучких підходах до автоматизації, які дозволяють системам швидко адаптуватися до змінних умов. Одним з ключових напрямків розв'язання цієї проблеми є розробка реконфігурованих мобільних роботів.

Реконфігуровані мобільні роботи представляють собою сучасні автоматизовані системи, які можуть змінювати свою конфігурацію або функціональність для виконання різних завдань в різних виробничих умовах. Це дає їм здатність ефективно пристосовуватися до змін в виробничому середовищі та вимогам завдань, що забезпечує підвищену гнучкість та оптимальне використання ресурсів.

У світі розвиток науки і техніки стоїть першому місці. Сьогодні однією з головних цілей сталого розвитку для України є підвищення ролі цифрових технологій у розвитку сфери навчання, оскільки від підростаючого покоління залежить майбутнє нашої країни. Освітня робототехніка стає дуже цікавою сферою знань величезній кількості молоді.

Для навчання учнів робототехніки використовують готові платформи – леґо-конструктори, з деталей якого можна легко створити програмованого робота. Недоліком є те, що елементи конструктора спрощені, усі мікроелектронні елементи модуля заховані за корпусом, мова програмування спеціально розроблена під конкретну модель конструктора.

Актуальність вибраної теми полягає в розробці реконфігурованого мобільного робота відкритої архітектури з набором різних модулів для навчання студентів на основі колісного робота з керуванням зі смартфона, з можливістю його зміни або оновлення додаванням різних модулів. Такий робот може бути використаний у навчальних закладах для вивчення основ робототехніки та програмування.

Об'єкт дослідження – процес керування реконфігурованим мобільним роботом.

Предмет дослідження – методи, конструкції, алгоритми та програмні забезпечення системи керування реконфігурованим мобільним роботом.

Мета роботи – підвищення ефективності проведення освітнього процесу за рахунок уніфікації модулів реконфігурованого мобільного робота.

Методи дослідження – методи об'єктно-орієнтованого програмування, теорія алгоритмів, теорія автоматичного керування, теорія схемотехніки.

Для досягнення поставленої мети необхідно вирішити такі задачі як:

- провести аналіз існуючих на даний момент реконфігурованих мобільних роботів, їх конструкції та системи керування;
- розробити структурну схему та схему електричну принципову спроектованого реконфігурованого мобільного робота;
- провести аналіз та обрати апаратні модулі для розробки реконфігурованого мобільного робота;
- розробити алгоритм роботи системи керування;
- розробити систему управління зібраним реконфігурованим мобільним роботом;
- зібрати макет реконфігурованого мобільного робота та провести експеримент;
- оформити пояснювальну записку згідно з ДСТУ 3008:2015 [1] та кваліфікаційну роботу, керуючись методичними вказівками [2].

1 АНАЛІЗ КОНСТРУКЦІЙ РЕКОНФІГУРОВАНИХ МОБІЛЬНИХ РОБОТІВ

1.1 Аналіз конструкцій реконфігурованих мобільних роботів.

Найбільш поширеним й цікавим напрямом робототехніки є конструювання, що дозволяє самостійно виконувати проектні роботи є комплекти конструкторів. Проектний підхід з використанням наборів LEGO дає учням можливість самостійно аналізувати "поле" творчої діяльності; самостійно намічати цілі й задачі майбутньої роботи; самостійно розробляти плани поетапного досягнення поставленої мети; здійснювати самоконтроль і самокорекцію; координувати свою діяльність з діяльністю інших.

Дослідження можливостей застосування Mindstorms – лінійки продуктів компанії Lego, що надає необхідні інструменти для створення простих роботів за допомогою кубиків Lego.

Mindstorms побудований на основі програмованого мікрокомп'ютерного блоку, який може керувати трьома двигунами і зчитувати з чотирьох датчиків. Двигуни та датчики можна підключити до блока управління за допомогою з'єднувальних роз'ємів, тому не потрібні спеціальні навички для складання функціонального набору електроніки робота. Сам програмований блок можна запрограмувати за допомогою графічної мови програмування NXT-G, яка йде в комплекті з наборами. Альтернативно, блок може бути запрограмований з використанням однієї з багатьох доступних сторонніх програм, які забезпечують підтримку таких мов, як C++ чи Java. Для складання конструктивних частин робота використовують кубики в стилі Lego Technic. Через це можна легко вставити блоки з інших наборів Lego для розширення потенційного рівня складності роботів.

Простота використання, розмір спільноти Mindstorms та знайомство з кубиками Lego робить його популярним вибором як платформи для

робототехніки в освіті, особливо при роботі з дітьми молодшого віку. Популярність Mindstorms виражається безліччю ресурсів, доступних для викладачів, таких як безліч книг, інтернет-спільноти та семінари, побудовані на основі екосистеми Mindstorms (рис. 1.1).

Одним із головних недоліків Lego Mindstorms є вартість. Інша проблема полягає в тому, що програмований блок обмежений максимум 3 двигунами та 4 датчиками. При додаванні ще одного двигуна або датчика до роботи стикаєшся з фізичною межею того, що можуть зробити роботи Mindstorms. Також складніше інтегрувати компоненти робототехніки сторонніх виробників у роботі Mindstorms.



Рисунок 1.1 – Lego Mindstorms NXT

Система Vex Robotics схожа на Lego Mindstorms у тому, що це платформа, яка надає всі необхідні елементи, необхідні створення функціонального робота: структурні компоненти, електроніка, програмне забезпечення. Основна відмінність полягає в тому, що, хоча Mindstorms має тенденцію бути схожою на іграшку, підходить для дітей молодшого віку. Натомість Vex вибирає серйозніший підхід до робототехніки, орієнтований на учнів старших класів та дорослих.

В якості елементів конструкції використовуються пластикові та перфоровані металеві балки. З'єднання виконуються за допомогою гайок і

гвинтів замість замикань і фрикційних з'єднань. Vex пропонує два різні варіанти мікроконтролерів для використання з її продуктами: мікроконтролер PIC і мікроконтролер Cortex, обидва з яких можна запрограмувати з використанням письмового коду, на відміну від графічної мови програмування, що використовується в Mindstorms.

Однією з основних вимог до функціонального робота є електронна система, яка може зчитувати датчики, обробляти інформацію та керуючі виходи.

Розроблена в 2005 році платформа Arduino являла собою комбінацію плати мікроконтролера, набір бібліотек C++ та інтегроване середовище розробки (IDE), спрямовану на створення мікроконтролерів, доступних художникам та дизайнерам.

Плати Arduino не призначені спеціально для використання у робототехніці, але цю функціональність можна додати за рахунок використання додаткових модулів. Ці модулі можна прикріпити поверх плати Arduino і забезпечте її додатковими функціональними можливостями, такими як дисплей або схема прямого керування постійним струмом двигуна.

Наявність великої кількості модулів та плат, простота використання, особливо в порівнянні з іншими мікроконтролерами, та низька вартість сприяли створенню Екосистеми Ардуїно (рис. 1.2).



Рисунок 1.2 – Arduino Leonardo

Часто роботи програмуються за допомогою низькорівневої текстової програми – мови програмування, наприклад, C. Ці текстові мови можуть бути досить складним завданням для людей без попереднього досвіду. Важко перевести концепції людської мови в алгоритми. Потрібно подбати про те, щоб синтаксис програми був правильним. Графічна мова програмування може полегшити проблему, одночасно забезпечуючи інтерфейс, який привабливіший для дітей. Scratch, розроблений у MIT Media Lab у 2006 році, є графічною мовою програмування, призначеною для навчання дітей принципам програмування у вигляді створення простих ігор та інтерактивного кіно. Blockly можна розглядати як набір бібліотек, які значно полегшують розробку Scratch-подібної мови програмування (рис. 1.3). Blockly написано на Javascript і призначено для роботи в середовищі веб-браузера. Використовуючи блочний інтерфейс прикладного програмування (API), можна легко визначити свій власний набір блоків, щоб створити повністю індивідуальну графічну мову програмування. Відмінною особливістю Blockly є те, що він може автоматично перевести програму Blockly в письмовий код, що читається. Мовні генератори для Javascript та Python вже існують. Проте з допомогою API можна створювати власні генератори.

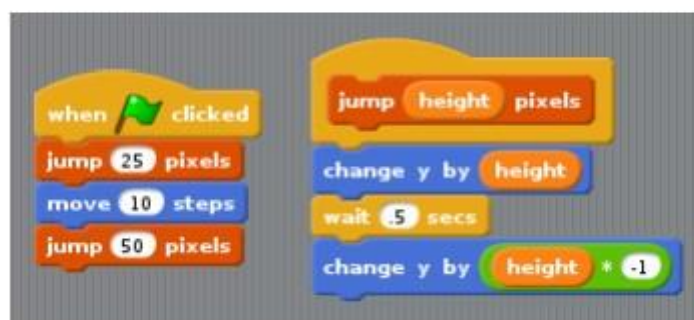


Рисунок 1.3 – Scratch

ArduBlock – це плагін для Arduino. Він є інтегрованим інструментом, який дозволяє писати програми для Arduino, використовуючи той самий стиль графічного оформлення блоків як Scratch та Blockly (рис. 1.4). Крім блоків, які

мають дослівний переклад функцій бібліотеки Arduino, це також надає деякі зумовлені блоки для роботи зі сторонніми компонентами Arduino, таких як реле або джойстик. Це чудовий інструмент для використання в освіті, особливо коли студенти вже познайомилися з аналогічною мовою, такими як Scratch.



Рисунок 1.4 – Blockly

Виробники роботів-аматорів, як правило, покладаються на широкий спектр методів, коли справа доходить до побудови фізичного втілення своїх роботів.

MakeBlock – це система комерційного виробництва, спеціально націленого на створення роботів. Система MakeBlock (рис. 1.5) побудована на основі декількох різних типів алюмінієвих балок, які з'єднуються разом за допомогою стандартних гвинтів.

Система MakeBlock – чудовий спосіб зробити дуже жорсткого робота. Різьбові отвори дозволяють легко підключати сторонні компоненти до рами. Система використовує один і той же отвір. Відстань між отворами, як у кубиків Lego Technic для покращення сумісності. Основні недоліки MakeBlock – це відносно висока вартість та обмеженість доступності.

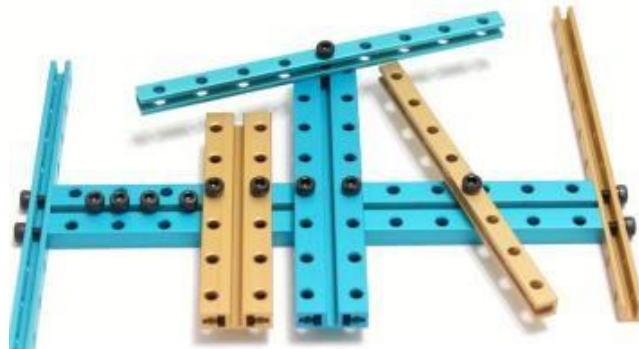


Рисунок 1.5 – MakeBlock

Новий спосіб навчання – методика DIY, обіцяє перетворити освіту зі спостереження за учнями в їх активну участь за допомогою оригінальних інтерактивних практичних уроків під керівництвом інструкторів та доповнений прикладами та обладнанням, які можна виготовляти лише за необхідності, що дозволяє персоналізувати та налаштовувати класи та індивідуальне навчання у загальних рамках.

Категорії, які були розглянуті, є комплексні робототехнічні платформи, електроніка, апаратне та програмне забезпечення. Взагалі, є два способи побудувати робота: або використовуючи повну робототехнічну платформу або створивши робота з нуля. Повні системи простіше у використанні, дозволяють досягти більш швидких результатів та краще підходять для молодих студентів. Недоліком є те, що вони, як правило, дорожчі. Навпаки, створення робота з нуля це набагато складніше і більше підходить для старшокласників, але дає набагато краще розуміння технології, є більш гнучким і може бути дешевшим. В останні роки створення робота з нуля стало набагато простіше завдяки численним проектам та продуктам, які реалізують певні аспекти робота, такі як апаратне забезпечення, програмне забезпечення або електроніка. Ці продукти та проекти часто можуть бути пов'язані з недавніми тенденціями DIY та Maker Movement.

Ці тенденції характеризуються використанням верстатів із ЧПУ та співробітництво через Інтернет для створення фізичних апаратних проектів. Можна зробити висновок, що DIY і Maker субкультура може вплинути на освіту, оскільки вона не тільки заохочує інтерес молодих людей до STEM пов'язаних областей, це також сприяє творчості та технологічному володінню. Всі ці навички, безсумнівно, матимуть завтра життєве важливе значення у суспільстві.

1.2 Аналіз аналогічних рішень

Одним із першопрохідників у світі модульної робототехніки став молодий інженер Джон Романишин. У 2013 році він створив схему для самоскладання

роботів. Невеликі кубики без зовнішніх рухомих частин можуть пересуватися, стрибати один на одного та з'єднуватися. Технологія одержала назву M-Blocks (рис. 1.6).



Рисунок 1.6 – M-Block

Через два роки він продемонстрував свою розробку на Міжнародній конференції IROS, довівши тим самим працездатність M-Blocks. Куби можуть рухатися без зовнішніх рухомих елементів. У середині кожного встановлено маховик, що обертається зі швидкістю 20 000 обертів на хвилину. У момент різкої зупинки "створюється" інерція, що спонукає куб рухатися у певному напрямку.

Щоб куби стикувалися між собою, на ребрах кожного є циліндричні магніти. Коли два куби зближуються, магніти повертаються один до одного різними полюсами. Коли куби далеко один від одного, вони пересуваються майже хаотично: стрибають у різні боки та роблять сальто. Натомість, поблизу працюють за принципом «ковзного куба» (рис. 1.7). Так називається ефект, коли один куб рухається вздовж іншого навколо загальної осі – намагніченого ребра.

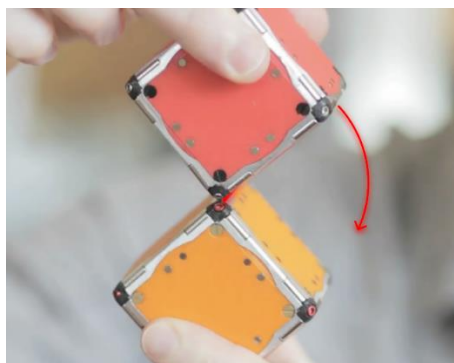


Рисунок 1.7 – Принцип «ковзного куба»

У першій версії M-Blocks куби пересувалися майже хаотично, за допомогою динаміки та магнітного впливу. Їм не вистачало «усвідомленості».

За шість років вийшла друга версія проекту. Куби-роботи «навчилися» спілкуватися один з одним, використовуючи систему, що нагадує штрих-код. На кожній із граней є свій код для ідентифікації. Також у кожному блоці встановлені датчики для зчитування напрямків та світлових сигналів. Так вони можуть розпізнавати один одного та збиратися у потрібні структури.

Сьогодні проект M-Blocks підтримують Національний науковий фонд та Amazon Robotics. Є надія на подальшу мініатюризацію пристрою. Одна з цілей проектів, подібних до M-Blocks, – створити систему мікроботів, здатну збиратися в складні структури.

Поки що реальним здається використання модульних роботів M-Blocks, наприклад, в інтерактивних іграх.

Сферичні роботи [22] ще один виходець Міжнародної конференції IEEE/RSJ з інтелектуальних робіт (IROS) – проект вчених з університету Шеньчжень. У 2020 році вони представили модульний робот FreeBOT (рис. 1.8).



Рисунок 1.8 – FreeBOT [22]

Візуально – повна протилежність M-Blocks. Модулі складаються із залізних сфер, усередині яких два колеса, моторчик, магніт, плата та батарейка (рис. 1.9).

За допомогою коліс виникає обертання та рух сфери по поверхні. Подібний пристрій є у відомого робота BB-8 виробництва Sphero.

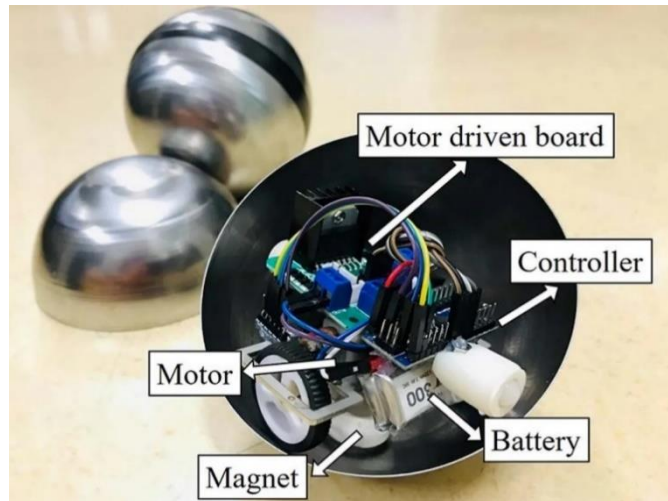


Рисунок 1.9 – Будова сфери M-Blocks [22]

Магніт спрямовує конструкцію та дозволяє з'єднувати сфери між собою. Завдяки феромагнітному корпусу сфери сильніше прилипають одна до одної. Магнітом'які властивості залізного корпусу дозволяють сфер легко роз'єднуватися. Достатньо відвести магніт в інший бік (рис. 1.10).

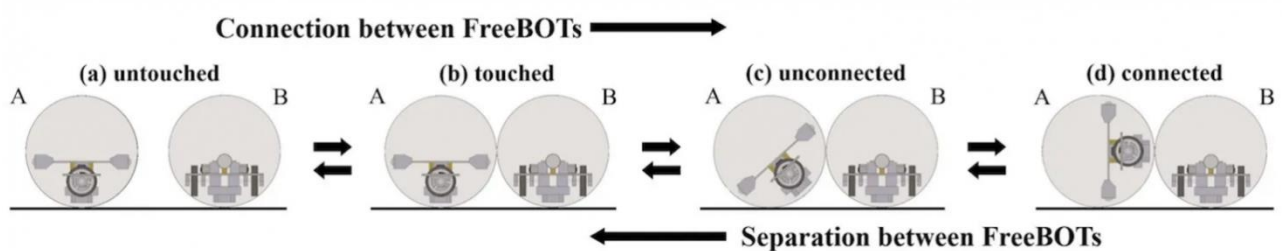


Рисунок 1.10 – Взаємодія сфер FreeBOT [22]

Модульний робот FreeBOT отримав продовження у самостійному проекті FreeSN. Це той самий модульний робот, але з іншою концепцією: тепер є «керуючі» та «дочірні» вузли. «Керуючі» вузли, можна сказати, займаються

перестановками «дочірніх» (рис. 1.11). Не потрібно розміщувати повноцінний механізм усередині кожної із сфер.

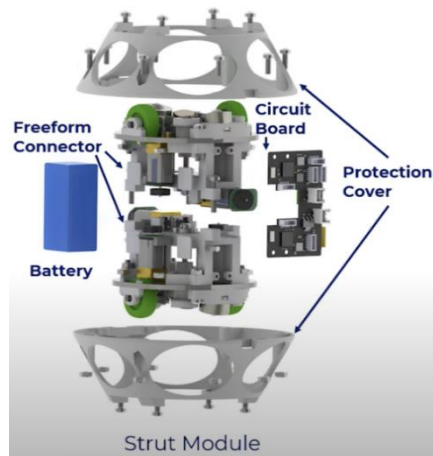


Рисунок 1.11 – Пристрій «керуючого» модуля [22]

Сьогодні FreeSN може втілювати різні структури – мости, триноги та ін. Також робот вміє переміщати важкі об'єкти.

Модульний робот із равликів [23].

Окрім FreeSN автори розробили новий самостійний проект SnailBot (рис. 1.12). Орієнтиром для нього стали равлики.

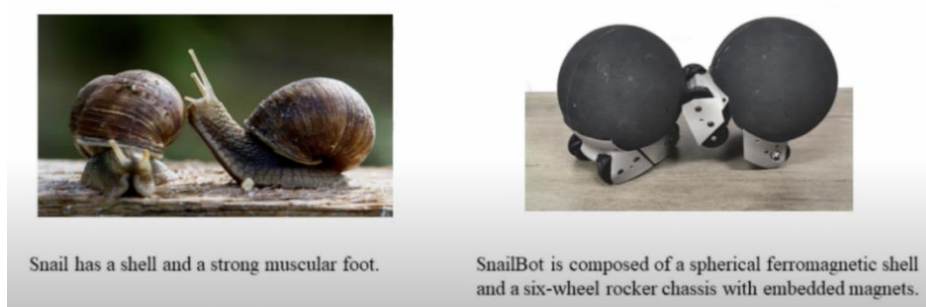


Рисунок 1.12 – SnailBot [23]

Модулі поєднуються в структури за допомогою магнітів. Але були додані шасі з шістьма колесами на підставі корпусу (рис. 1.13).

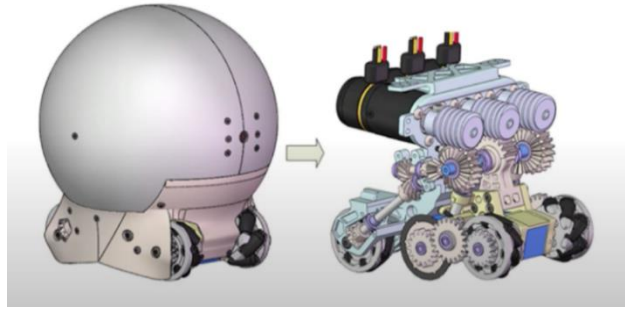


Рисунок 1.13 – CAD-модель SnailBot [23]

Завдяки зовнішній конструкції робот став мобільнішим. Автори стверджують, що такий механізм може підвищити ефективність перебудови структури та зменшити кількість збоїв під час стикування вузлів.

Mori. Робот-орігамі [24].

Модульні роботи можуть створювати різні структури. Головна їхня властивість – здатність до реконфігурації, тобто до перебудови складання.

Усі попередні претенденти на звання «реконфігурованого робота» були громіздкими в порівнянні з роботом-орігамі Mori. Основа цього робота – трикутні модулі. Проект було засновано дослідниками з Політехнічної Школи Лозанни у 2016 році.

За допомогою трикутних модулів можна збирати структури щільного компонування – як і у випадку M-Blocks, мінімальні зазори (рис. 1.14).

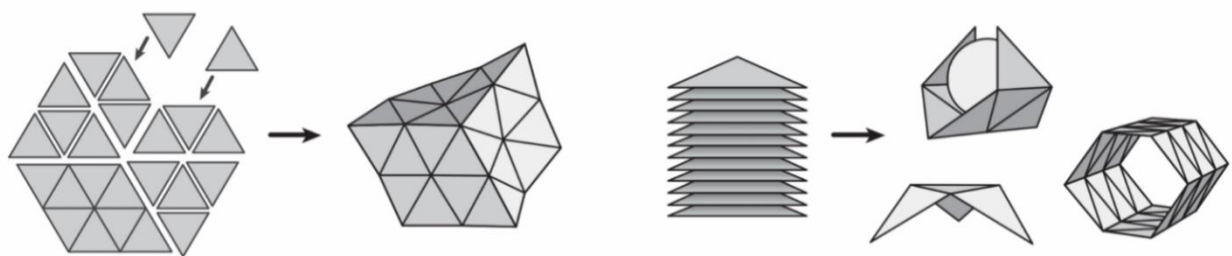


Рисунок 1.14 – Структури [24]

Перспективи орігамі: завдяки структурі та товщині роботів-орігамі можна створювати різні двовимірні «ґрати». Потенційно їх можна застосувати у

розумній «броні». Властивості оригамі дозволяють повторювати складні геометричні форми.

Також можна знайти їм застосування у космонавтиці. Наприклад, у реконфігурованих супутникових батареях (рис. 1.15). Тоді космічний сміття не зможе знеструмити космічний апарат.

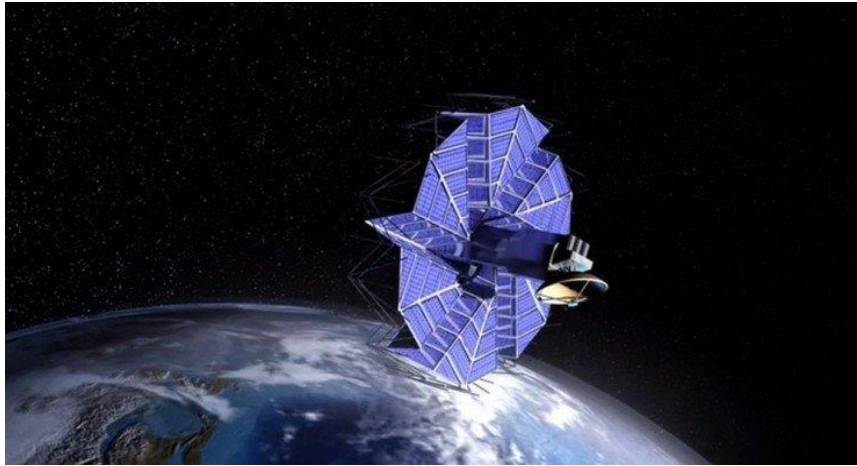


Рисунок 1.15 – Супутникові батареї [24]

Як і в FreeSN, робот-оригамі складається з «дочірніх» та «керуючих» вузлів (рис. 1.16). Останні займаються перестановками, крізь них відбувається управління.



Рисунок 1.16 – Структура оригамі-робота [24]

Довжина кожного з модулів становить 70 мм, товщина – 6 мм, а вага – 26 г (рис. 1.17). Це оптимальні параметри для реконфігурацій. В M-Blocks система "керуючі-дочірні" неможлива через масу вузлів.

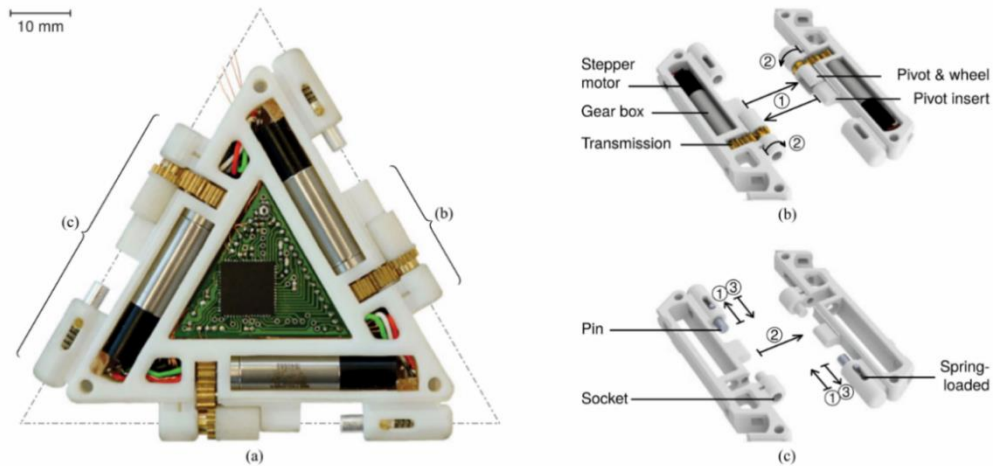


Рисунок 1.17 – Mori, пристрій модуля [24]

Через плату відбувається керування моторчиками та шарнірами. За допомогою них робот може переміщатися поверхнею. Автори не приховують, що для переміщення потрібна рівна поверхня. Це перший мінус.

Другий мінус – стикування з іншими вузлами неможливе без участі людини. Також кожен із модулів управляється окремо. Автори ще не реалізували міжмодульну сигналізацію, коли можна керувати всією структурою через один «несучий» модуль.

Третій мінус – відсутність внутрішнього джерела живлення. До кожного з «керуючих» модулів протягнуто дроти для живлення самої плати. У роботів M-Blocks, FreeBOT, FreeSN та SnailBot використовують внутрішні джерела живлення. Але питання можливості заряджати їх без руйнування структури залишається відкритим.

1.3 Аналіз методів керування реконфігурованим мобільним роботом

Класифікацію систем автоматичного управління (САУ) можна провести за методом керування та функціональною ознакою (рис. 1.18).

За методом керування всі системи поділяються на дві великі категорії: звичайні (несамонастроювальні) та адаптивні (самонастроювальні).

Звичайні системи, які належать до категорії простих, не змінюють свою структуру в процесі керування. Вони досить добре розроблені і широко використовуються в ливарних та термічних цехах. Звичайні САУ можна поділити на три підкласи: розкриті, закриті та комбіновані системи керування.

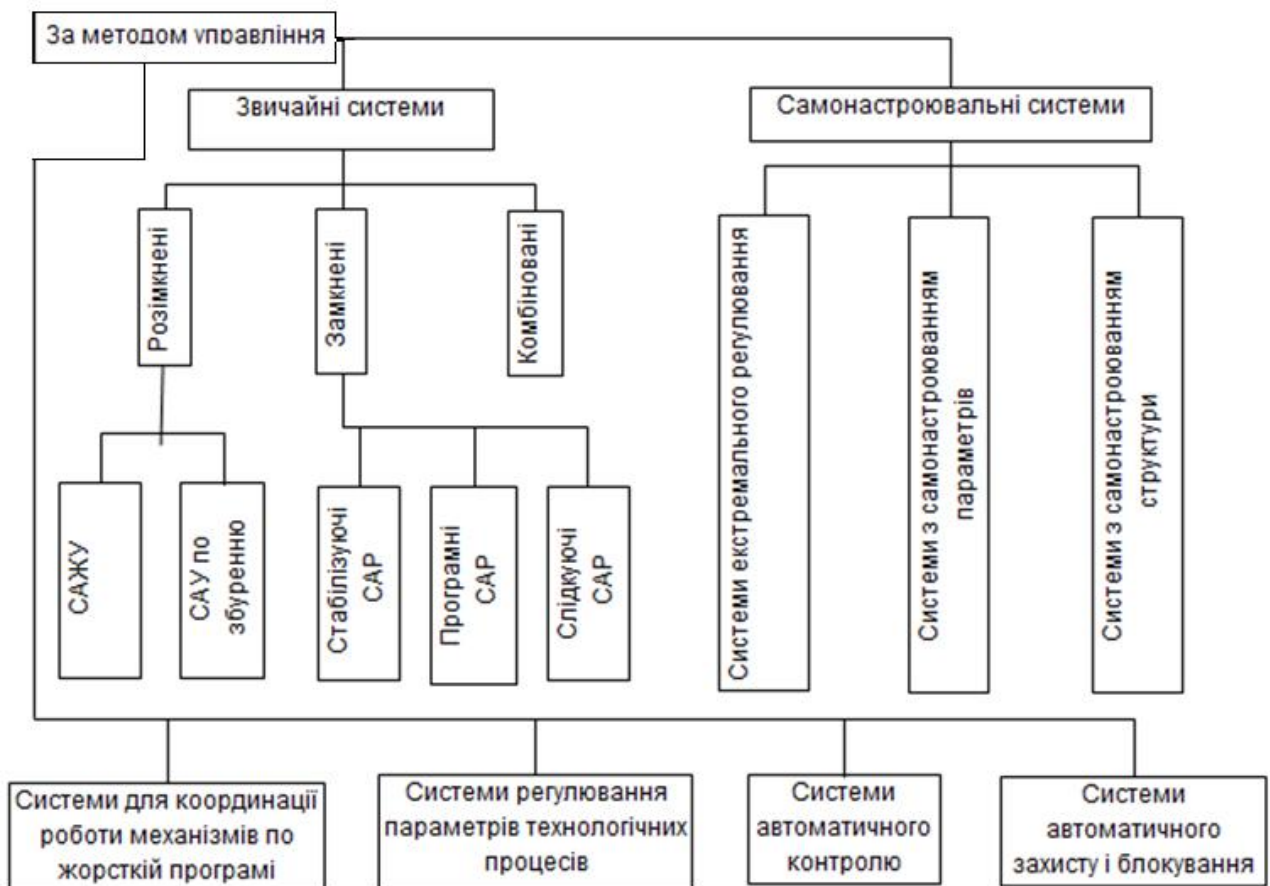
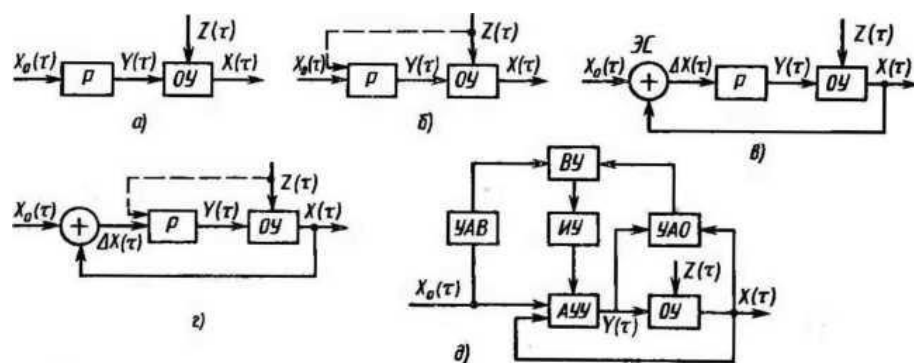


Рисунок 1.18 – Класифікація систем автоматичного управління

Розкриті САУ, у свою чергу, поділяються на системи автоматичного жорсткого керування (САЖК) та системи керування за збуренням.

У перших системах регулятор P (рис. 1.19, а) впливає на об'єкт управління OY незалежно від отриманого результату, тобто значення керованої величини $X(t)$ і зовнішнього збурення $1(t)$. Системи керування за збуренням (рис. 1.19, б) працюють за принципом, коли керуючий вплив $Y(t)$ формується залежно від зовнішнього збурення $1(t)$, яке впливає на об'єкт керування. Наприклад, система опалення ливарного або термічного цеху може бути прикладом такої системи. У цьому випадку споживання гарячої води в тепловій мережі цеху залежить від зовнішніх погодних умов. Чим холодніше на вулиці, тим більше гарячої води подається до опалювальних батарей, і навпаки.

Замкнуті системи автоматичного керування, що працюють за принципом відхилення, також називають системами автоматичного регулювання (САР). Однією з їх відмінних особливостей є наявність замкнутого контуру проходження сигналів, тобто наявність зворотного зв'язку, через який інформація про стан керованої величини $X(t)$ передається на вхід порівняльного елемента ПЕ (рис. 1.19, в). Системи автоматичного регулювання призначені для вирішення трьох завдань: стабілізації керованої величини (стабілізуюча САР), зміни керованої величини за відомою (програмованою САР) або невідомою (слідкуюча САР) програмами.



- а – замкнена; б – незамкнена за відхиленням; в – замкнута;
 г – комбінована; д – що самонастроюється;
 P – регулятор; OY – об'єкт управління; ПЕ – порівняльний елемент;
 ПАВ – пристрій аналізу впливу, що задається;
 ОП – обчислювальний пристрій; ВП – виконавчий пристрій;
 АУУ – автоматичний управляючий пристрій;
 УОТ – пристрій аналізу об'єкту управління

Рисунок 1.19 – Схеми керування [12]

.У стабілізуючих САР задане значення керованої величини постійне. Прикладом такої системи може слугувати система регулювання температури в робочому просторі термічної печі. У програмових САР значення керованої величини змінюється у часі заздалегідь розробленою (відомою) програмою. У слідкуючих системах задане значення керованої величини змінюється у часі за наперед не відомою програмою. Слідкуючі та програмові САР відрізняються від стабілізуючих принципом обробки задаючого сигналу. Найтипівішим прикладом слідкуючого регулювання є автоматичне підтримання заданого співвідношення між витратами палива і повітря при регулюванні процесу горіння у паливних плавильних і нагрівальних печах. Комбіновані системи (рис. 1.19, г) поєднують у собі переваги систем керування за відхиленням і за збуренням, що підвищує точність керування. Дія незважених збурень у комбінованих системах компенсується або послаблюється керуванням за відхиленням.

Самонастроювальні (адаптивні) системи (рис. 1.19, д) можна розділити на три підкласи: екстремальні системи, системи з самонастроюванням параметрів і системи з самонастроюванням структури.

Екстремальними системами регулювання називають системи стабілізуючого, слідкуючого або програмного керування, у яких налаштування, програма або закон відтворення автоматично змінюються залежно від зміни зовнішніх умов або внутрішнього стану системи з метою створення найвигіднішого (оптимального) режиму роботи об'єкта управління. У таких системах замість постійного налаштування або програми встановлюється пристрій автоматичного пошуку, який проводить аналіз деякої характеристики об'єкта (коефіцієнта корисної дії, продуктивності, економічності, тощо) і залежно від отриманого результату подає в керуючий пристрій потрібне значення керованої величини так, щоб дана характеристика отримала екстремальне значення при неперервному зміні різних збурювальних впливів, що впливають на умови роботи системи.

У системах з самонастроюванням параметрів при зміні зовнішніх умов або характеристик об'єкта регулювання відбувається автоматична (не за заздалегідь заданою програмою) зміна змінних параметрів керуючого пристрою з метою забезпечення стійкої роботи системи і підтримки регульованої величини на заданому або оптимальному рівні.

У системах з самонастроюванням структури при зміні зовнішніх умов та характеристик об'єкта управління відбувається перемикання елементів у схемі з'єднань або введення в неї нових елементів. Метою таких змін (відбору) структури є досягнення кращого рішення задачі управління. Відбір структури здійснюється шляхом автоматичного пошуку з використанням обчислювальних і логічних операцій. Такі системи повинні не тільки адаптуватись до всіх змін зовнішніх умов та характеристик об'єкта, але й нормально функціонувати навіть при відмові або несправності окремих елементів, створюючи нові ланцюжки замість пошкоджених. Системи з самонастроюванням структури можна змусити самовдосконалюватися, "набувати досвід" шляхом швидкого випробування кількох варіантів, відбору та "запам'ятовування" найкращого з них.

Згідно з класифікацією за функціональним ознакою, всі автоматичні системи управління поділяються на чотири класи: системи для координації роботи механізмів; системи регулювання параметрів технологічних процесів; системи автоматичного контролю; системи автоматичного захисту і блокування. Системи, призначені для координації роботи окремих механізмів установки або установки в цілому, є системами автоматичного жорсткого управління (САЖУ). Системи автоматичного регулювання (САР) технологічних процесів забезпечують підтримку регульованої величини на заданому рівні або зміну її за заданою програмою. Системи автоматичного контролю (САК) містять засоби та методи отримання інформації про поточні значення параметрів технологічних процесів (температури, тиску, забрудненості або забрудненості повітря та ін.) без прямої участі людини. Системи автоматичного захисту (САЗ) і блокування

(САБ) запобігають виникненню аварійних ситуацій у роботі обладнання при встановленому режимі.

Система керування, яку я обрав, має схожий принцип роботи з системою автоматичного регулювання.

1.4 Висновки до першого розділу

В результаті дослідження визначені продукти, які можна використовувати як інструменти для реалізації проектів робототехніки в освіті. Огляд технологій, які можна використовувати для впровадження робототехніки до освітнього закладу, наведений у публікації (Додаток Б).

Створення роботів – популярний вибір проекту для впровадження проблемно-орієнтованого навчання (PBL) у навчальних закладах. Причиною, чому це такий популярний вибір, пояснюється багатопрофільністю теми: робототехніка вимагає безлічі різних наукових, технічних і технологічних навичок, як фізика, електроніка, математика та програмування.

Однією з ключових переваг модульної архітектури реконфігурованого мобільного робота є простота заміни модулів. Завдяки стандартним з'єднувачам, модулі можуть бути легко вилучені та замінені іншими, що дозволяє швидко адаптувати робота до різних завдань. За допомогою робота можна проводити дослідження принципів роботи бездротового зв'язку та набувати практики роботи з сервоприводами та створення роботизованих маніпуляторів. Для вивчення робототехніки бажано починати з платформ Arduino. Arduino відома тим, що має велику кількість різних датчиків, модулів та величезну кількість матеріалу щодо їх підключення до Arduino.

У розділі обґрунтовано нормативно-методичну основу формування системи управління автономним роботом. Визначено необхідність використання створення такої системи. Проведено аналіз схожих систем. Вибрана система автоматичного керування реконфігурованим мобільним роботом.

2 РОЗРОБКА СТРУКТУРИ ТА СИСТЕМИ КЕРУВАННЯ РЕКОНФИГУРОВАНИМ МОБІЛЬНИМ РОБОТОМ

2.1 Аналіз апаратних модулів

Основні апаратні модулі пристрою це мікроконтролер Arduino Uno Rev3 (ATmega16U2); шилд підключення датчиків для Arduino Uno Rev3; акумулятор, що перезаряджається, із зарядним модулем TP4056 Mini-USB з функцією захисту акумулятора.

Arduino Uno Rev3 (ATmega16U2) (рис. 2.1) це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega16u2.



Рисунок 2.1 – Arduino Uno Rev3 (ATmega16U2)

Контролер відрізняється мікросхемою USB-UART перехідника (ATmega16u2), яка має високу швидкість передачі даних і для якої не потрібна додаткова установка драйверів – драйвера встановлюються автоматично при встановленні середовища розробки Arduino IDE, додатковими контактами SDA та SCL (I2C інтерфейс) та виводами AREF джерела опорної напруги для АЦП контролера та IOREF – виходом напруги живлення портів введення-виведення (для автоматичного перемикання напруги периферії при використанні 5 В та 3,3 В контролерів). Мікроконтролер ATmega16u2 дозволяє використовувати

Сенсор-шилд для Arduino UNO (рис. 2.3) призначений для збільшення кількості контактних груп і дозволяє підключити велику кількість різноманітних модулів: сенсори, сервоприводи, реле, кнопки, потенціометри та ін. У новій версії окремо виведені роз'єми для підключення аналогових датчиків, а також роз'єми: UART, I2C, SPI, APC220 для стандартних Bluetooth-модулів та ультразвуковий інтерфейс URF01.

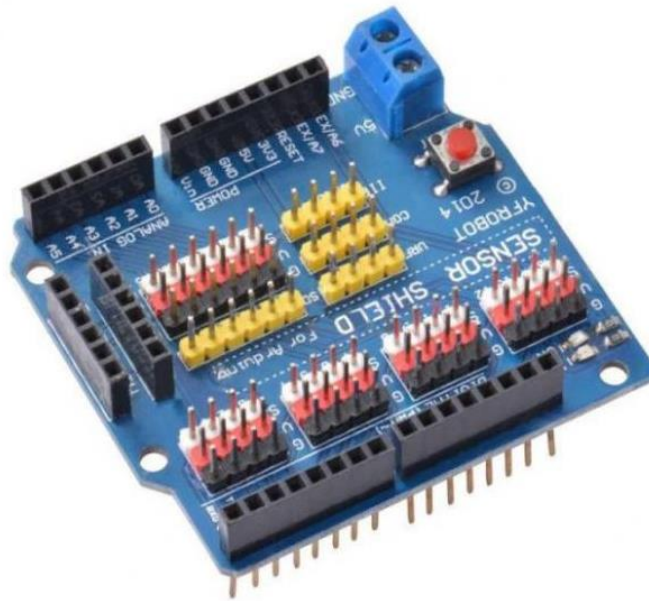


Рисунок 2.3 – Шилд підключення датчиків для Arduino UNO R3

На платі передбачено роз'єм для використання зовнішнього джерела живлення (тільки 5 В). Також на всі основні контакти проведено лінію живлення. Додатково є кнопка скидання та індикатор живлення. Для зручного сприйняття крім стандартних позначень на платі модуля, виконано кольорове маркування контактів.

Енергопостачання включає в себе зарядний модуль TP4056 Micro-USB (рис. 2.4) з функцією захисту акумулятора, два акумулятори LG M58T 18650 Li-Ion 2200 мАг, батарейний відсік для двох акумуляторів 18650 з кришкою та фланцями для встановлення в корпус пристрою.

Зарядний пристрій TP4056 Micro-USB використовується для заряджання Li-ion батарей та контролю їх розрядки, заряджає до напруги 4,2 В струмом 1 А.

Можна регулювати струм заряду (за допомогою заміни резистора R3). Мікросхема TP4056 захищає батарею від перерозряду (коли напруга падає нижче 2,4 В). Також модуль містить елементи захисту від переполюсування та струмовий захист. Вхідна напруга може подаватися як за допомогою підпаяних проводів, так і за допомогою мікро-USB роз'єму.



Рисунок 2.4 – Зарядний пристрій TP4056 Micro-USB

2.2 Структура модульного реконфігуруємого робота

2.2.1 Шасі

Рухома частина являє собою 4-х колісну двопалубну повнопривідну робо-платформу (рис. 2.5) з можливістю розширення та дообладнання.

На платформу кріпляться всі модулі. Платформа оснащена енкодерами на кожному колесі. Розміри платформи 235 мм x 155 мм.

Кожне колесо підключено до свого двигуна постійного струму з редуктором. Мікродвигун постійного струму встановлений в пластмасовий корпус, у якому знаходиться редуктор з пластмасових шестерінок, що знижує швидкість обертання та збільшує зусилля на валу механізму. На вал мотор-редуктора насаджується колесо з гумовою шиною. Таким чином, задаючи різні швидкості обертання цих двигунів, реалізується рух робота у різних напрямках.

В автономному режимі здійснюється рух уперед, назад, а також виконуються повороти на місці (по і проти годинникової стрілки).

Технічні характеристики двигуна:

- споживаний струм: 70 мА (250 мА максимум) за напруги 3 В;
- напруга живлення: від 3 В до 12 В (рекомендоване 6 В);
- крутний момент: 800 г/см (за напруги 3 В);
- передавальне число редуктора: 1:48;
- швидкість обертання без навантаження: 314 об/хв (за напруги 3 В);
- діаметр колеса: 66 мм;
- швидкість: 48 м/хв;
- вага двигуна: 50 г;
- розміри двигуна: 70 мм x 22 мм x 18 мм.

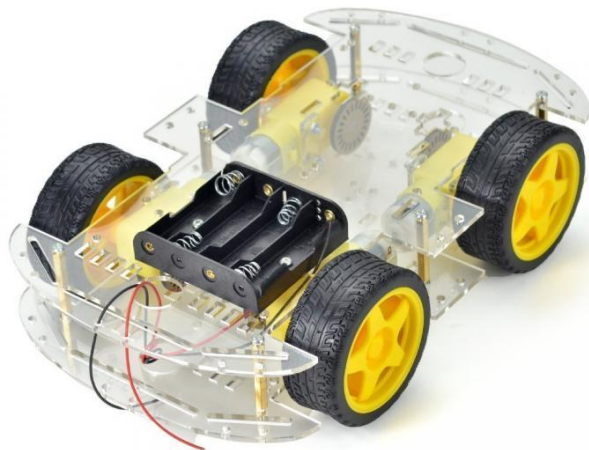


Рисунок 2.5 – Робо-платформа 4-х колісна двопалубна повнопривідна

2.2.2 Модуль руху

Модуль являє собою драйвер двигунів на базі чіпа L298.

Драйвер двигунів L298N керує як швидкістю, так і напрямом обертання електродвигуна постійного струму. Він використовує систему L298N PWM, яка може керувати напругою прямокутних імпульсів. Чим ширші імпульси, тим

швидше обертатиметься двигун. Однак точна ширина імпульсу залежатиме від типу двигуна, яким він керує. ІС драйвера двигуна L298 з подвійним повним мостом з високим струмом та напругою, в основному розроблена для того, щоб дозволити типовим рівням логіки TTL керувати різними індуктивними навантаженнями, такими як двигуни постійного струму, соленоїди, реле, крокові двигуни тощо. буд. Драйвер може управляти двома моторами або одним кроковим двофазним двигуном. Підтримує роботу з керуючим мікроконтролером із напругою рівнів 3,3 В. Керуюче живлення для двигунів VMS: 5~35 В, сила струму 2 А на міст. Перемичка "5V enable" служить для вибору джерела живлення чіпа L298: на модулі встановлений стабілізатор живлення, який обрізає вхідну напругу до +5 В і подає її на чіп.

Технічні характеристики драйвера двигунів:

- напруга живлення вбудованої логіки: 5 В;
- споживаний струм вбудованої логіки: 0 – 36 мА;
- напруга живлення драйвера: 5 – 35 В (максимальне 46 В);
- робочий струм драйвера: 2 А (піковий струм 3 А);
- максимальне споживання енергії: 25 Вт;
- робоча температура: -20 °С - +135 °С;
- габарити: 43,5 мм x 43,2 мм x 29,4 мм;
- вага: 26 г.

Модуль руху (рис. 2.6) призначений для управління рухом вперед/назад і поворотами для вивчення базових алгоритмів керування.

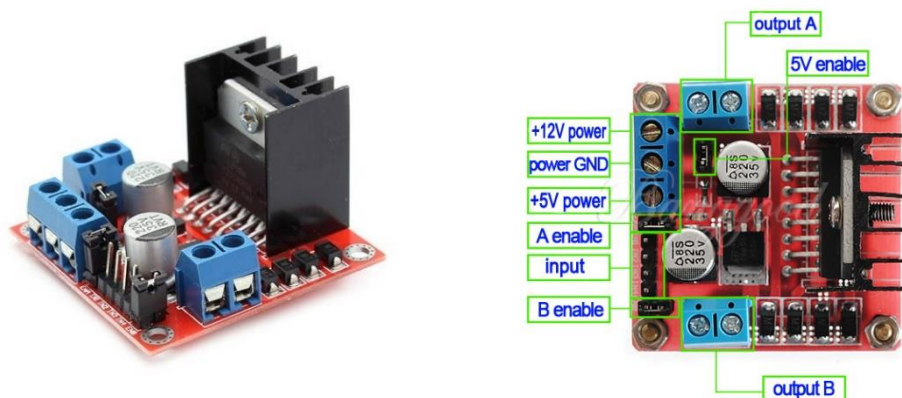


Рисунок 2.6 – Драйвер двох двигунів на L298N

2.2.3 Модуль керування

Модуль HC-05 – це простий у використанні модуль Bluetooth SPP (Serial Port Protocol) (рис. 2.7) призначений для прозорого налаштування бездротового послідовного з'єднання. Модуль Bluetooth із послідовним портом повністю відповідає стандарту Bluetooth V2.0+EDR (розширена швидкість передачі даних) 3 Мбіт/с із повним радіоприймачем 2,4 ГГц і базовою смугою. Він використовує CSR Bluecore 04-зовнішню одночипову систему Bluetooth з технологією CMOS і AFH (функцією адаптивного перемикавання частоти). Модуль HC-05 працює при низькій напрузі 3,0 В і елементах управління введенням/виведенням від 3,0 В до 4,2 В. Він має вбудовану антену, крайовий роз'єм та інтерфейс UART із програмованою швидкістю передачі даних.

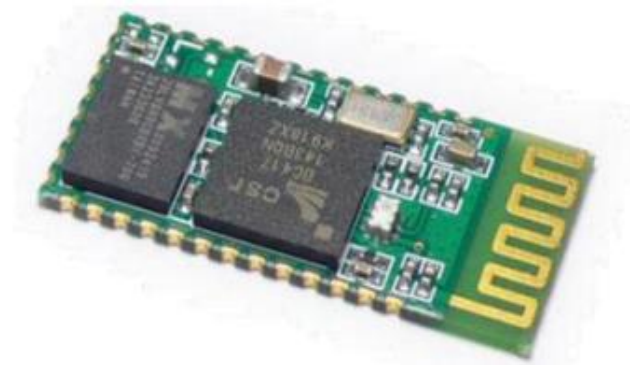


Рисунок 2.7 – Модуль HC-05 [6]

Технічні характеристики Bluetooth модуля HC-05:

- протокол зв'язку Bluetooth Specification v2.0+EDR;
- частота GFSK (Gaussian Frequency Shift Keying);
- потужність відправки ≤ 4 дБм, Class 2;
- потужність прийому ≤ -84 дБм (at 0,1% BER;
- швидкість асинхронна 2,1 Мбіт/с (Max) / 160 кбіт/с, синхронна 1 Мбіт/с / 1 Мбіт/с;
- безпека Authentication and encryption;
- профіль Bluetooth serial port;

- живлення +5 В DC 50 мА;
- робочі температури -20 °С ~ +75 °С;
- розміри: 26,9 мм × 13 мм × 2,2 мм.

2.2.4 Модуль лінії (F233-01)

Інфрачервоний 4-канальний датчик перешкод F233-01 (рис. 2.8) – це надійний модуль для мікроконтролерних платформ Arduino та інших, призначений для виявлення перешкод, прямування лініями та автоматизації. Пристрій оснащений інфрачервоними випромінювачами та фотоприймачами, що забезпечує високу точність та ефективність у робототехнічних проектах. Регульована дальність виявлення та простота підключення роблять цей модуль універсальним вибором для широкого спектра завдань.



Рисунок 2.8 – 4-канальний інфрачервоний модуль Arduino

Інфрачервоний випромінювач створює сигнал, який відбивається від об'єктів. Інтенсивність відбитого світла визначається кольором та відстанню об'єкта. Червоний індикатор спалахує при перешкоді, а поріг чутливості можна налаштувати за допомогою потенціометра.

Датчик перешкод використовується для створення пристроїв реакції та об'їзду перешкод. Кількість напрямків – 4.

Детектує перешкоди на відстані 10 мм - 60 см (може налаштовуватися).

Технічні характеристики модуля об'їзду перешкод:

- каналів детектування – 4, настройка чутливості кожного – окрема;
- робоча напруга: DC 3,3 В – 5 В;
- робочий струм: 1 А;
- робоча температура: - 10 °С - + 50 °С;
- монтажний отвір: М3;
- розміри модуля: 42 мм × 38 мм × 12 мм;
- розміри зонду: 25 мм × 12 мм × 12 мм;
- вага: 22 г.

2.2.5 Модуль захвату предметів

Модуль захвату предметів (рис. 2.9) призначений для формування навичок роботи з технічними пристроями, вмінню маніпулювати предметами, отримання досвіду поєднання навчання при створенні універсальної роботизованої руки з програмуванням на платформі Arduino, навчання управлінню сервомоторами.



Рисунок 2.9 – Механічний маніпулятор для Arduino SG90 [16]

Маніпуляторна рука-робот 4-DOF ступенів свободи на базі 4-х сервомоторів типу sg90 та плати керування Arduino Uno. Це аналог сучасних промислових роботів, створена передусім для учнів середніх шкіл, студентів вузів чи технічних гуртків для ознайомлення з елементами прикладної механіки, робототехніки чи мікропроцесорної техніки.

Після підключення зібраного механізму до комп'ютера через USB (рис. 2.10) можна відправляти команди через серійний порт для управління сервоприводами. У той же час, управління можливе не лише через Arduino, а і Raspberry Pi або інші контролери. Також можливі власні процедури запису та здійснення фіксованих повторень. Керування рукою у всіх напрямках може здійснюватися шляхом 4-х потенціометрів або пультом дистанційного керування.

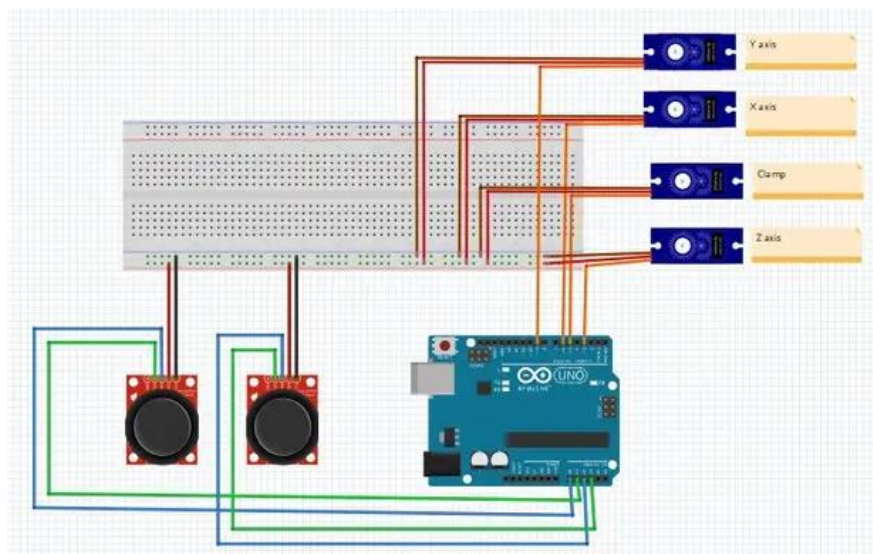


Рисунок 2.10 – Схема підключення механічного маніпулятора

Механічний маніпулятор включає чотири серводвигуни SG90, Для керування маніпулятором використаний контролер Arduino Uno R3 на процесорі Atmega328P.

2.2.6 Серводвигуни SG90

Це електромеханічні системи, які широко використовуються в техніці, як приводи для точного прямолінійного або кутового руху. Мікросервопривод SG90 використовується для керування рухом різних механізмів та елементів роботів, таких як кермо, камери, хвати тощо (рис. 2.11).

Технічні характеристики мікросервоприводу SG90:

- швидкість без навантаження: 0,12 с / 60 °С при живленні 4,8 В;
- крутний момент: 2 кг/см;
- температурний діапазон: 0 °С +50 °С;
- ширина мертвої зони: 4 мікросекунди;
- робоча напруга живлення: 3,5 – 5 В;
- споживаний струм у русі: 50-80 мА;
- споживаний струм утриманні: 5-10 мА;
- кут повороту 120°;
- розміри: 3,3 см x 3 см x 1,3 см;
- вага: 9 г.



Рисунок 2.11 – Основні компоненти сервомотора [16]

Двигун постійного струму перетворює електричну напругу, що подається на його клеми, в обертальний рух. Коробка передач зменшує швидкість, яку забезпечує двигун постійного струму, і збільшує крутний момент, прикладений

до вихідного валу. Електронна плата керування зчитує ШІМ-сигнал, що надсилається на сервопривід, перетворює його на кут, порівнює його з поточним кутом і обчислює керуючий сигнал. Професійні сервомотори зазвичай мають енкодер як датчик положення. Сервоприводи з підключеним енкодером передають користувачеві поточне положення.

2.2.7 Модуль освітлення

Призначений для реалізації ефектів висвітлення та для реакції на події (наприклад, сигнал для виявлення перешкоди).

Реалізований колом з 12 RGB світлодіодами WS2812 з послідовним керуванням (рис. 2.12).



Рисунок 2.12 – Коло з 12 RGB світлодіодами WS2812 з послідовним керуванням

Технічні характеристики інфрачервоного модуля:

- кількість світлодіодів: 12;
- напруга живлення: 5 В постійного струму;
- модель світлодіодів: WS2812 RGB;
- інтерфейс: послідовний однопровідний;
- зовнішній діаметр: 50 мм;
- внутрішній діаметр: 35 мм;
- вага: 6 г.

2.4.8 Модуль відображення інформації LCD 1602 I2C

Призначений для виведення інформації про стан робота (наприклад, рівень батареї, дані датчиків).

Символьний синій ж/к дисплей LCD 1602 з адаптером I2C (рис. 2.13) дає можливість відтворювати значення і дані датчиків безпосередньо на LCD дисплей, а не на екран комп'ютера. Дисплей має 2 рядки по 16 символів, в його пам'яті можна записати до 8-и довільних символів користувача. Дисплей LCD 1602 з I2C адаптером підключається до контролера по двох дротах, на контролері також є регулювання яскравості підсвічування (змінний резистор).



Рисунок 2.13 – Модуль відображення інформації LCD 1602 I2C

LCD дисплей 1602 обладнаний 4-х піновим роз'ємом стандарту 2,54 мм.

SCL: послідовна лінія тактування (Serial CLock).

SDA: послідовна лінія даних (Serial DAta).

VCC: "+" живлення.

GND: "-" живлення.

Технічні характеристики модуля відображення інформації:

- розміри 80 мм x 36 мм;
- робоча температура 0~50 °С;
- підсвічування блакитне;

- колір символів білий;
- розмір символу: 4,35 мм x 2,95 мм;
- формат 16 x 2;
- розміри крапки 0,5 мм x 0,5 мм;
- інтерфейс I2C;
- видима область 64,5 мм x 13,8 мм;
- живлення 5 В.

2.2.10 Модуль датчика вологості та температури DHT11

Модуль DHT11 – цифровий датчик температури та вологості призначений для вимірювання температури і вологості повітря (рис. 2.14). Передача даних здійснюється по одному дроту з використанням власного протоколу. Може бути використаний в пристроях на Arduino, AVR, PIC, ARM та ін.

Одна з найважливіших функцій полягає в тому, що температура і вологість вимірюються з точністю до десятих часток; тобто до одного десятичного знака. Але, враховуючи його продуктивність і ціну, цим можна знехтувати.

Датчик містить в собі АЦП для перетворення аналогових значень вологості та температури.

Корпус складається із двох частин. Усередині корпусу на стороні датчиків знаходяться датчик вологості та датчик температури NTC (термістор).

Чутливий до вологості компонент, який використовується для вимірювання вологості, має два електроди з вологоутримуючою підкладкою (зазвичай сіль або провідний пластиковий полімер), затиснутої між ними. між електродами (рис. 2.15). Зміна опору між двома електродами пропорційно відносною. Більш висока відносна вологість зменшує опір між електродами, у той час як нижча відносна вологість збільшує цей опір.

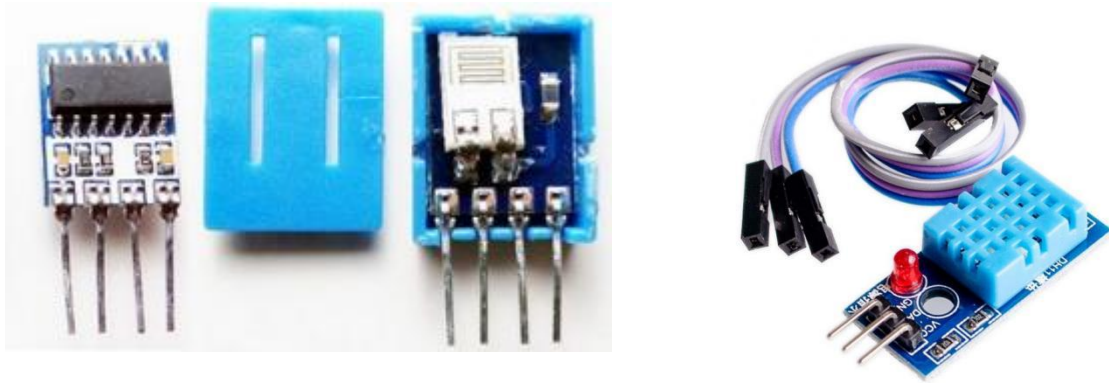


Рисунок 2.14 – Модуль датчика вологості та температури DHT11

Технічні характеристики датчика DHT11:

- напруга живлення: 5 В;
- діапазон температур: 0 – 50 °С;
- похибка показань температури: ± 2 °С;
- діапазон вологості: 20 – 90%;
- похибка вологості: $\pm 5\%$.

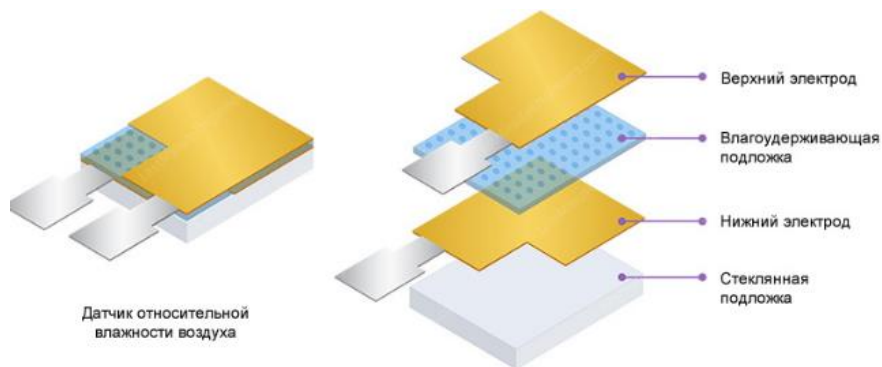


Рисунок 2.15 – Внутрішня структура датчика вологості DHT11

Крім того, в цьому датчику вимірювання температури є датчик температури NTC (термістор). Термістор це терморезистор, тобто резистор, який змінює свій опір в залежності від температури. Технічно всі резистори є термісторами – їх опір трохи змінюється в залежності від температури, але зазвичай ця зміна дуже мало і її важко виміряти. Термістори зроблені так, щоб їхній опір різко змінювався при зміні температури, і зміна на один градус може становити

100 Ом або більше (рис. 2.16). Термін "NTC" означає "Negative Temperature Coefficient" (негативний температурний коефіцієнт), що означає, що зі зростанням температури опір зменшується.

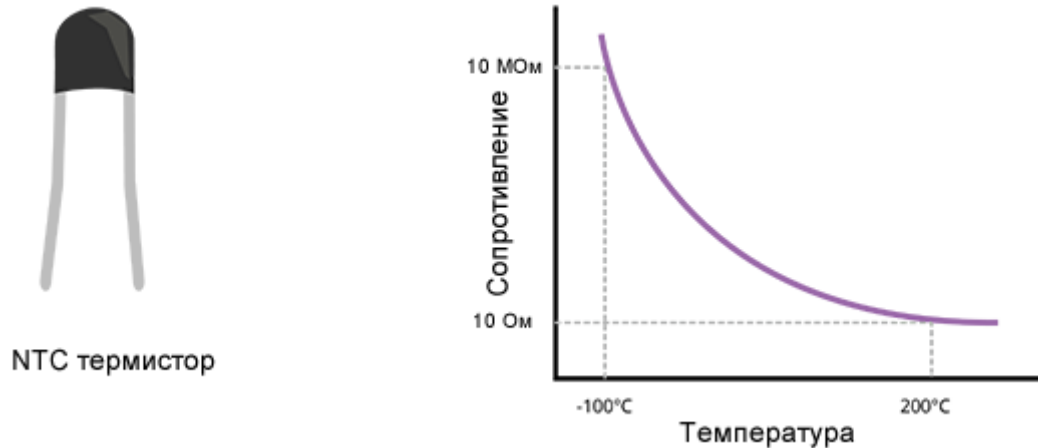


Рисунок 2.16 – Графік залежності опору термістора NTC від температури

Проведемо експеримент: будемо зчитувати дані з датчика DHT11 і виводити на екран LCD дисплею 1602. Рекомендована схема підключення Arduino містить обов'язковий для однопровідних ліній резистор-підтяжку до VCC, в якості опції рекомендується конденсатор (фільтр живлення між VCC і GND). У нашому випадку DHT11 у вигляді модуля, його можна підключати до Arduino безпосередньо – резистор і конденсатор там вже є. Схема підключення датчика DHT11 та WH1602 до плати Arduino показана на рисунку 2.17. Для живлення ПКІ потрібен окремий блок живлення +5 В.

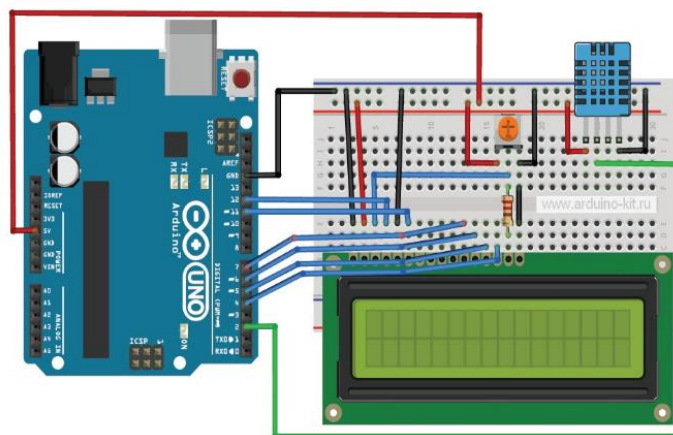


Рисунок 2.17 – Схема підключення DHT11 до плати Arduino

Напишемо скетч. Для роботи з датчиками DHT11 в Arduino є бібліотека OneWire. Вміст скетчу для читання даних з датчика та виведення на екран індикатора WH1602 показано у лістингу.

```
#include "DHT.h"
#define DHTPIN 2 // пін підключення контакту DATA
#define DHTTYPE DHT11 // DHT 11
#include <LiquidCrystal.h>
// ініціалізація із зазначенням контактів підключення
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
  lcd.begin(16,2); // режим роботи
  dht.begin();
}
void loop()
{
  // отримання з датчика даних вологості та температури
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(t) || isnan(h)) // помилка отримання даних
  {
    lcd.clear();lcd.setCursor(0,0);
    lcd.print("Failed to read");
  }
  else // вивести дані на РКІ
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Humidity: ");lcd.print(h); lcd.println(" %");
    lcd.setCursor(0,1);
    lcd.print("Temp: "); lcd.print(t);lcd.println(" *C");
  }
  delay(2000); // пауза перед наступним виміром
}
```

Для відображення даних температури і вологості змінюватимемо значення на дисплеї кожні 2 с. А оскільки значення обох показників не буде перевищувати

2 цифр, будемо підставляти на початок і кінець даних символи, що дозволяють їх ідентифікувати – t і h на початку та C і % в кінці.

2.3 Розробка загальної структурної схеми

Структурна схема представляє собою умовне графічне зображення сукупності частин об'єкта, або системи керування та зв'язків між цими частинами. Структурна схема для реконфігурованого мобільного робота та вхідних модулів, наведена на рисунку 2.18.

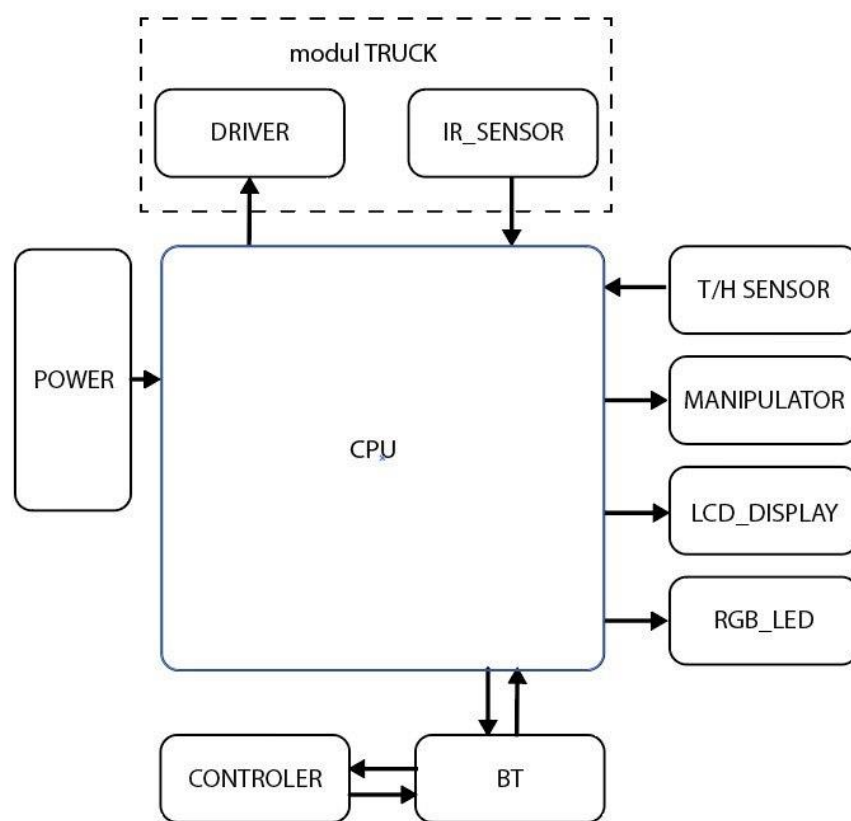


Рисунок 2.18 – Структурна схема системи керування реконфігурованим мобільним роботом

Функціональна схема представляє собою умовне графічне зображення, що відображає функції, які виконуються окремими елементами системи і зв'язки

між ними. На функціональній схемі (рис. 2.19) зображена загальна схема реконфігурованого мобільного робота та керуючих пристроїв.

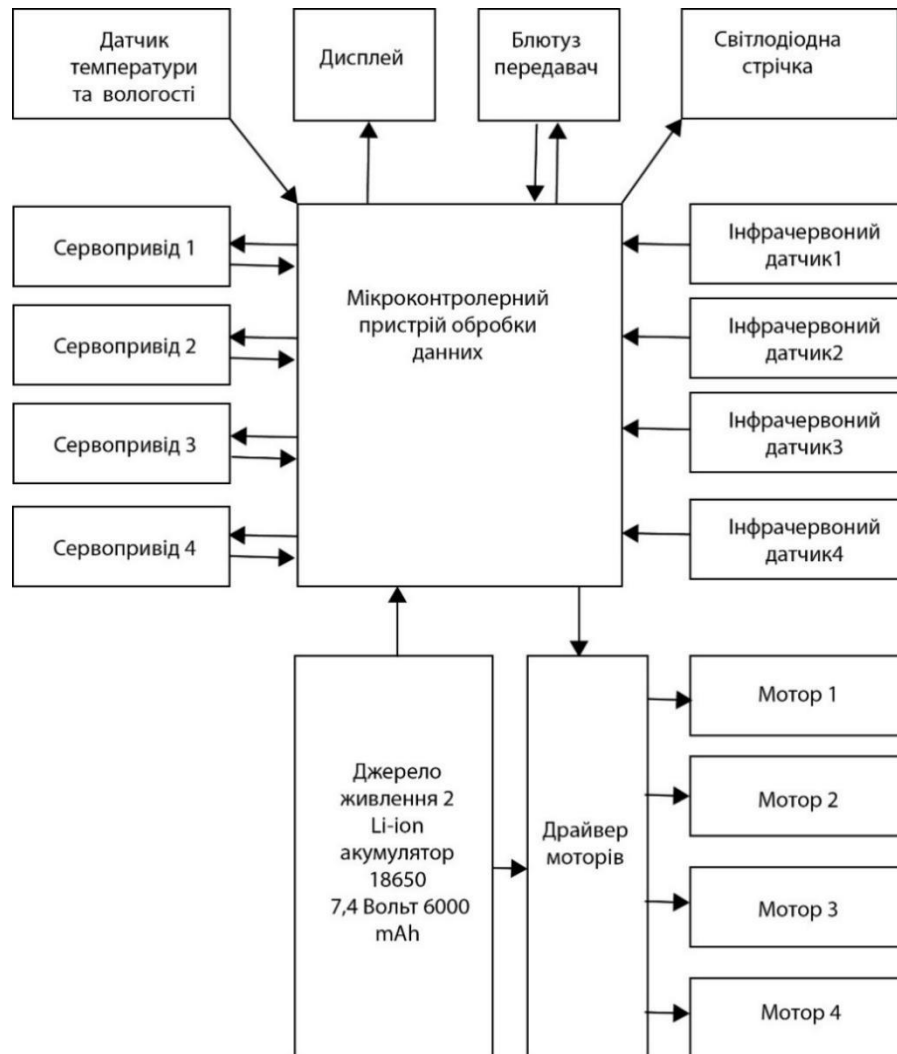


Рисунок 2.19 – Функціональна схема системи керування реконфігурованим мобільним роботом

Реконфігурований мобільний робот базується на мікроконтролері ATmega328p, який повністю відповідає технічним вимогам, таким як кількість входів та виходів для підключення елементів керування, таких як сервоприводи, bluetooth модуль.

Bluetooth передавач є засобом передачі даних зі смартфона до модулю руху та сервоприводів. Такий протокол був обраний через простоту у підключенні, стабільність роботи та енергоефективність у порівнянні з Wi-Fi протоколом.

2.4 Можливі схеми підключення

Схема підключення електричних компонентів зображена на рисунках 2.20-2.22.

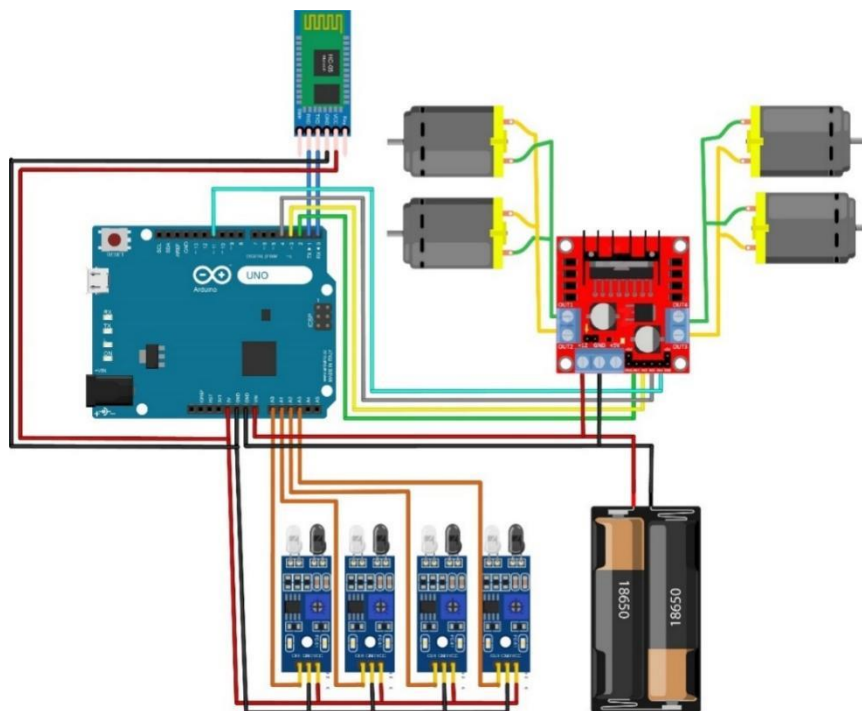


Рисунок 2.20 – Схема підключення апаратних модулів керуючої системи

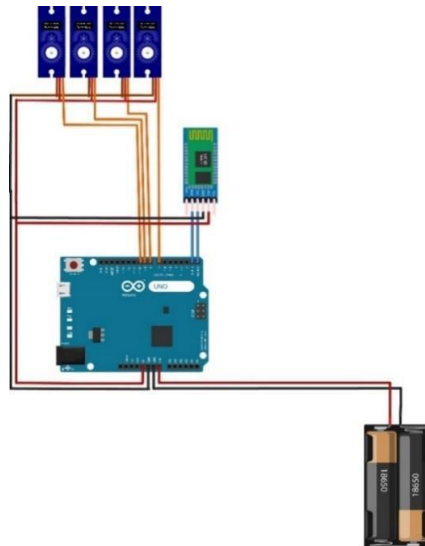


Рисунок 2.21 – Схема підключення механічного маніпулятора

Третя схема (рис. 2.22) передбачає підключення модуля відображення інформації LCD 1602 I2C, 4-канального інфрачервоного модуля Arduino, модуля датчика вологості та температури DHT11. Система управляється по Bluetooth. Датчик вологості та температури заміряє температуру та вологість навколишнього середовища, ця інформація виводиться на екран модуля відображення. Інфрачервоний модуль Arduino світиться.

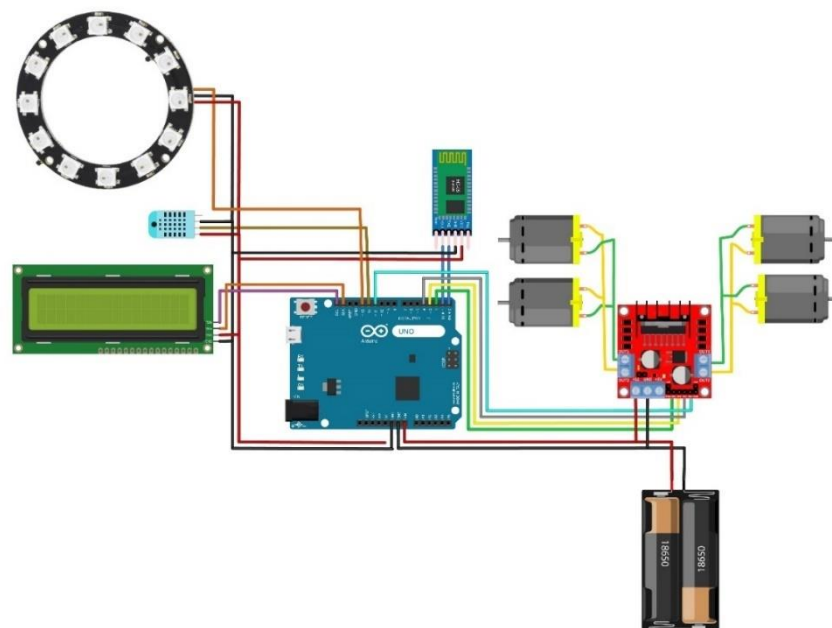


Рисунок 2.22 – Схема підключення додаткових модулів

2.5 Складання рівняння двигуна постійного струму з незалежним збудженням

Теорія автоматичного управління в широкому розумінні цього слова є теорія побудови і функціонування систем автоматичного управління.

САУ можна розділити на дві характерні групи, які відрізняються одна від одної як за принципом управління, так і за особливостями функціонування – розімкнуті САУ, що базуються на принципі керування за збуренням, і замкнуті САУ, які діють на основі принципу керування за відхиленням.

При відхиленні від стану рівноваги в надходженні і витратах енергії через дію збурень в системі виникає відхилення регульованої величини від заданого значення. Це призводить до включення в роботу регулятора, який намагається його зменшити, внаслідок цього виникає перехідний процес.

Вибір двигуна для приводу (двигуна з редуктором і, якщо потрібно, підсилювачем) різних автоматичних систем – один з основних етапів проектування системи. Це пояснюється тим, що двигун, з одного боку, обумовлює тип, параметри і потужність попередніх каскадів підсилювачів, а також потужність джерела енергії. З іншого боку, від двигуна в значній мірі залежать динамічні особливості системи, які він може забезпечити. Установлення двигунів надмірної потужності приводить до невиправданого зростання габаритних розмірів і маси приводу, погіршенню енергетичних показників і т. і. Двигуни, потужність яких нижче необхідної, не забезпечують рух вихідного вала за заданим законом або можуть перегрітися, отже, різко знижується строк служби приводу. При правильному виборі потужності двигуна температура його нагріву мало відрізняється від допустимої, а динамічні можливості приводу використовуються повністю.

Вихідною величиною буде швидкість обертання ω , яка регулюється зміною напруги U , що подається в якірне коло двигуна. Таким чином, треба встановити в динаміці залежність $\omega = f(u)$. Запишемо рівняння електричної рівноваги для

якірного кола двигуна, маючи на увазі, що в режимі двигуна напруга живлення обчислюється за формулою

$$u = L_{\text{я}} \frac{di_{\text{я}}}{dt} + R_{\text{я}} i_{\text{я}} + e, \quad (2.1)$$

де $L_{\text{я}}, R_{\text{я}}$ – індуктивність і активний опір кола якоря;

e – проти-ЕРС (електрорушійна сила) двигуна.

Для режиму розгону двигуна можна записати рівняння динаміки у вигляді

$$M_{\text{дин}} = M_{\text{дв}} - M_{\text{ст}},$$

де $M_{\text{дин}}, M_{\text{дв}}, M_{\text{ст}}$ – відповідно динамічний, двигунів і статичний моменти двигуна.

Коли статичний момент невеликий, тобто $M_{\text{ст}} \approx 0$, то можна записати

$$J \frac{d\omega}{dt} = c i_{\text{я}}.$$

Звідки

$$i_{\text{я}} = \frac{J}{c} \frac{d\omega}{dt},$$

де J – момент інерції двигуна;

c – стала струму.

Підставляючи знайдене значення $i_{\text{я}}$ в рівняння електричної рівноваги кола якоря двигуна, після деяких перетворень дістанемо

$$\frac{R_{я}L_{я}J}{R_{я}c} \frac{d^2\omega}{dt^2} + \frac{R_{я}J}{c} \frac{d\omega}{dt} + c_e\omega = u. \quad (2.2).$$

Введемо позначення:

$$\frac{L_{я}}{R_{я}} = T_{я},$$

$$\frac{R_{я}J}{cc_e} = T_M,$$

$$\frac{1}{c_e} = k_D,$$

$$T_{я} = \frac{0,028}{0,28} = 0,1c.$$

Коефіцієнт двигуна k_D визначимо за паспортними даними для номінального режиму у статиці:

$$c_e\omega = u - i_{я}R_{я}.$$

Звідки

$$c_e = \frac{u - i_{я}R_{я}}{\omega}, \quad (2.3)$$

$$k_D = \frac{1}{c_e} = \frac{1}{\frac{u - i_{я}R_{я}}{\omega}} = \frac{\omega}{u - i_{я}R_{я}} = \frac{\omega}{U_{ян} - I_{ян}R_{я}} = \frac{314}{186 - 43 \cdot 0,28} = \frac{314}{173,96} = 1,805 \frac{рад}{В \cdot c}.$$

При цьому рівняння динаміки двигуна постійного струму в операторній формі матиме вигляд

$$(T_{я}T_M p^2 + T_M p + 1)\omega = k_D u.$$

Та остаточно передатна функція двигуна буде

$$W(p) = \frac{k_D}{T_J T_M p^2 + T_J p + 1} \quad (2.4)$$

Якщо $T_M = \frac{R_J J}{c c_e} > 2T_J = 2\sqrt{\frac{L_J J}{c c_e}}$, або $R_J \sqrt{\frac{J}{c c_e}} > 2\sqrt{L_J}$, то двигун при даному співвідношенні параметрів буде аперіодичною ланкою другого порядку, а в протилежному випадку – коливальною ланкою.

Визначимо коефіцієнт T_M :

$$T_M = \frac{J R_J}{k^2},$$

$$\text{де } k = \frac{1}{k_D}.$$

Тоді

$$T_M = J R_J k_D^2 = 0,28 \cdot 0,28 \cdot 3,258 = 0,255 \text{ с.}$$

Таким чином, $T_M=0,255 > 2 \cdot T_J=0,2$, тобто двигун є аперіодичною ланкою другого порядку.

Аналіз передаточної функції двигуна постійного струму з незалежним збудженням може бути представлена послідовним з'єднанням аперіодичної ланки першого порядку і інтегруючої ланки, охопленої від'ємним зворотнім зв'язком по частоті обертання. Складемо структурну схему двигуна постійного струму (рис. 2.23).

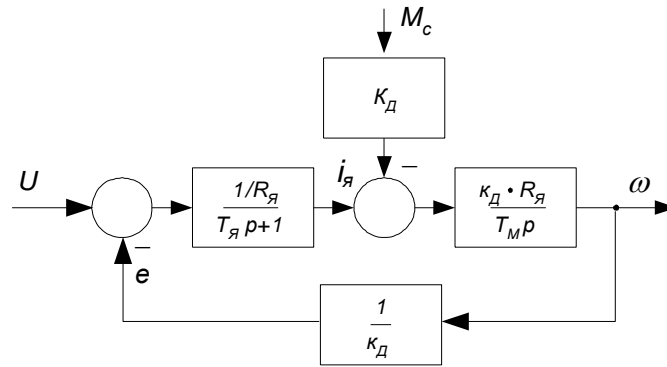


Рисунок 2.23 – Структурна схема двигуна постійного струму

З урахуванням наведеної структурної схеми двигуна отримуємо повний вираз для передаточної функції в залежності від обох зовнішніх впливів:

$$\omega = U_{\text{я}} \frac{\kappa_{\text{д}}}{T_{\text{м}} T_{\text{я}} p^2 + T_{\text{м}} p + 1} - M_{\text{с}} \frac{R_{\text{я}} \kappa_{\text{д}} (T_{\text{я}} p + 1)}{T_{\text{м}} T_{\text{я}} p^2 + T_{\text{м}} p + 1}. \quad (2.5)$$

Підставимо розраховані значення:

$$\omega = U_{\text{я}} \frac{1,805}{0,0255p^2 + 0,255p + 1} - M_{\text{с}} \frac{0,505 \cdot (0,1p + 1)}{0,0255p^2 + 0,255p + 1}.$$

Потрібно відмітити, що в статичному режимі (при $p = 0$) формула (2.5) співпадає з відомою формулою механічної характеристики ДПС:

$$\omega_{\text{ст}} = U_{\text{я}} \kappa_{\text{д}} - M_{\text{с}} R_{\text{я}} \kappa_{\text{д}}^2 = \frac{U_{\text{я}}}{C\Phi} - M_{\text{с}} \frac{R_{\text{я}}}{(C\Phi)^2},$$

$$\omega_{\text{ст}} = 1,805 \cdot U_{\text{я}} - 0,912 \cdot M_{\text{с}}.$$

2.6 Складання макету мобільного реконфігурованого робота

При проектуванні мобільного реконфігурованого робота модульної конструкції (рис. 2.24) основними завданнями було створити робота з матеріалів, що доступні та не є дорогими.

В роботі використано всі покупні деталі, крім пластин з пластику, на які кріпляться додаткові модулі. Завдяки цьому ціна всього робота вийшла відносно не великою, приблизно 3000 грн.

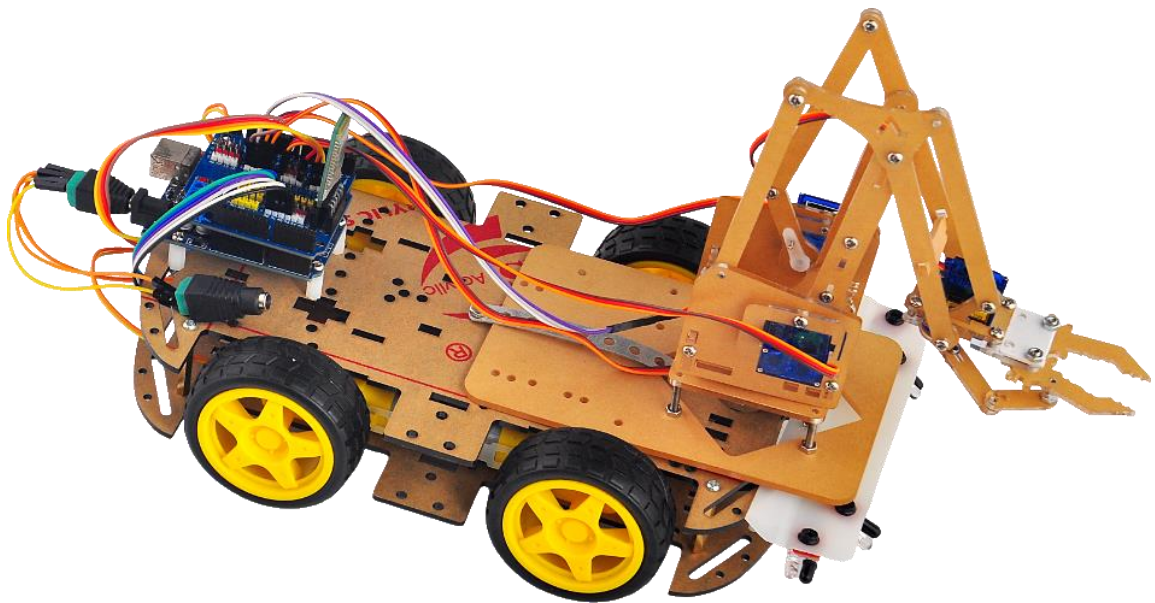


Рисунок 2.24 – Загальний вигляд макету модульного робота

Цей макет складається з модулів, наведених в частині 2.1, 2.2 цієї роботи:

- мікроконтролер Arduino Uno Rev3 (ATmega16U2);
- шилд підключення датчиків для Arduino Uno Rev3;
- акумулятор з зарядним модулем TP4056 Mini-USB з функцією захисту акумулятора;
- драйвер двигунів на базі чіпа L298;
- робо-платформа 4-х колісна двопалубна повнопривідна.

На рисунку 2.25 зображений цей макет у розкладеному стані.

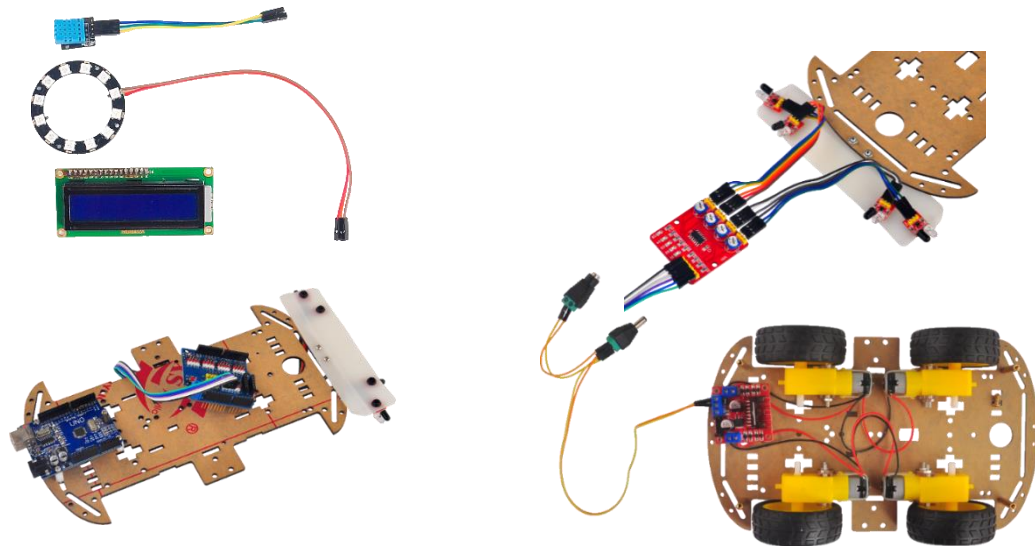


Рисунок 2.25 – Деталі макету модульного реконфігурованого робота

Процес складання можна умовно поділити на 2 етапи. Перший етап це складання робоплатформи, а другим етапом є встановлення додаткових модулів, що є основою реконфігурації робота.

На першому етапі збирається готове шасі. Підключаються мотори, до Arduino. Для початку роботи необхідно завантажити та встановити бібліотеку Adafruit AFMotor.

Приклад коду для керування двигунами:

```
#include <AFMotor.h> // підключити бібліотеку Adafruit
#include <Servo.h> // підключити бібліотеку для сервомотора
AF_DCMotor motor(1); // створити об'єкт мотор, вказавши номер роз'єму DC
мотора на пла
Servo servo; // створити об'єкт сервомотор
servo.attach(10); // приєднати серво на пін 9 або 10 (крайній роз'єм на платі M
motor.setSpeed(speed); // встановити швидкість DC мотора від 0 (зупинка) до
255 (повно
motor.run (RELEASE); // DC мотор стоп
motor.run (FORWARD); // DC двигун вперед
motor.run (BACKWARD); // DC двигун назад
```

```
servo.write(90); // повернути серво на 90 град.
```

Підключаємо 4-канальний інфрачервоний модуль Arduino. Далі підключаємо модуль Bluetooth HC-05. Після подачі живлення на модуль Bluetooth HC-05 на ньому повинен заморгати світлодіод, що означає працездатність блютузу. Включаємо Bluetooth на мобільному телефоні, знаходимо пристрій з ім'ям HC-05 та підключаємося.

Фотографія зібраної 4-х колісної двопалубної повноприводної робо-платформи (шасі) наведена на рисунку 2.26.



Рисунок 2.26 – Шасі у зібраному вигляді

Другий етап починається зі складання механічного маніпулятора, кріплення серводвигунів SG90 до основи маніпулятора. Механічний маніпулятор зображений на рисунку 2.27.

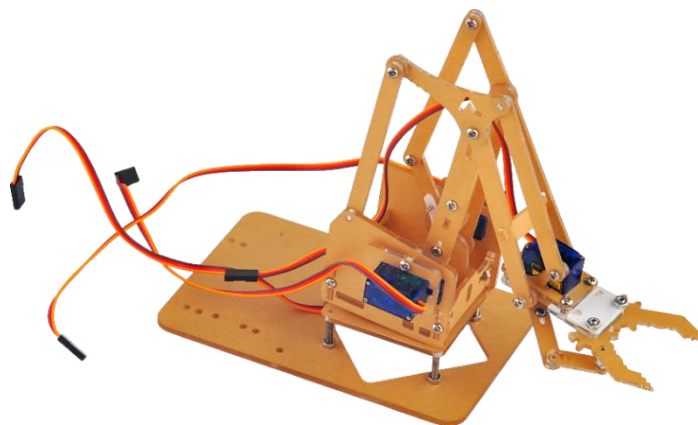


Рисунок 2.27 – Механічний маніпулятор у зібраному вигляді

Після завершення складання першого та другого етапу ці деталі поєднуються та під'єднуються до мікроконтролера з Bluetooth модулем та блоком живлення, і ми отримуємо модуль для експериментів – макет мобільного робота (рис. 2.28), до роз'ємів якого користувачі можуть підключати свої додаткові компоненти (датчики та модулі), створюючи унікальні рішення.

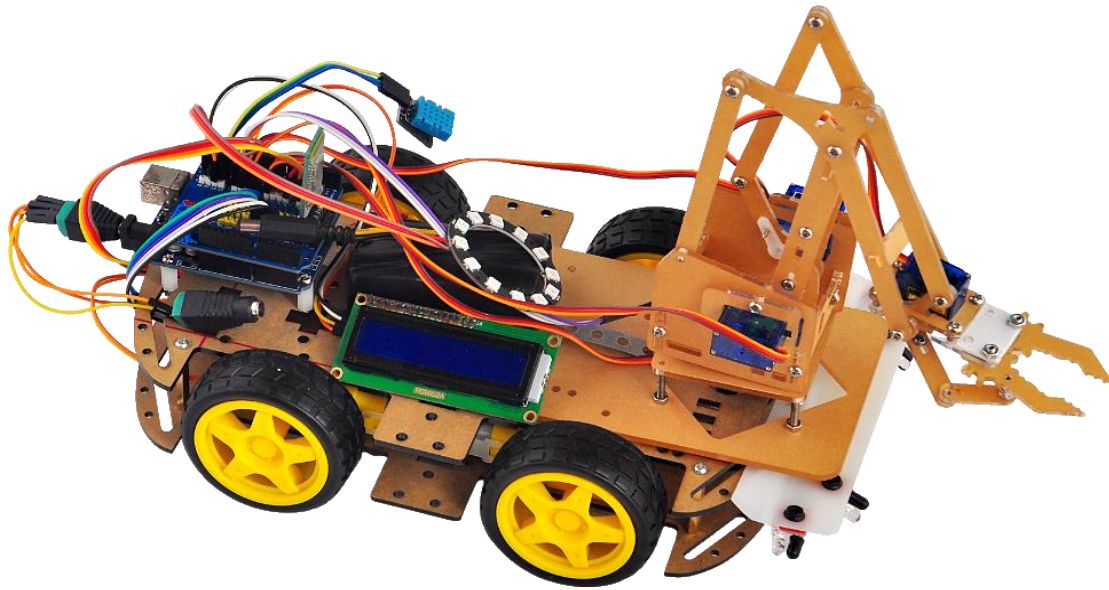


Рисунок 2.28 – Макет реконфігурованого мобільного робота у зборі

2.7 Висновки до другого розділу

У розділі був проведений аналіз апаратних модулів, вибрані всі елементи, що використовувалися під час складання реконфігурованого мобільного робота. Розроблені загальна структурна схема для мобільного робота та системи керування ним, а також функціональна схема, що відображає функції окремих елементів системи і зв'язки між ними. На основі розроблених схем підключення апаратних модулів керуючої системи та схеми підключення апаратних модулів реконфігурованого мобільного робота було проведено складання макету робота. Усі датчики були підключені до мікроконтролера, пов'язаного бездротовим зв'язком зі смартфоном. Напруга подається від блока живлення на плату. Потужності достатньо для нормального функціонування всіх модулів та

датчиків. Керування двигунами постійного струму організоване за допомогою ШІМ-регулятора.

Система керування мобільним роботом призначена для забезпечення спільної роботи всіх систем і компонентів робота з метою переміщення робочого органу у задану точку або виконання рухів за заданою траєкторією.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір середовища розробки

При розробці програмного продукту важливим чинником є правильний вибір засобів програмної реалізації, що впливає на час розробки, якість, надійність та швидкість кінцевого продукту. Також важливо враховувати особливості, можливості й поширення операційної системи, під керівництвом якої буде виконуватися програма.

У теперішній час у різноманітній техніці та системах автоматизації використовується мікроконтролери, вони є достатньо дешевими, малогабаритними та достатньо потужними для виконання задач з обробки даних знятих з датчиків та їх подальше використання. Досить широко відома така фірма виготовлення мікроконтролерних плат як Arduino. Ці плати базуються на мікроконтролерах Atmega. Вони є надійними, простими у застосуванні та дешевими, крім того вони легко перепрограмуються за рахунок вбудованого програматора, завдяки якому можна завантажувати код напряму з комп'ютера.

Для написання коду для мікроконтролерів існує багато середовищ розробки.

Мова програмування C++.

Раніше програми писалися мовою машин або ж машинним кодом, який складався з примітивних інструкцій, які могли бути прочитані лише машиною. Програми, що були написані машинним кодом, були важкими для розуміння, зазвичай тому що структура машинного коду відбивала дизайн комп'ютера а ніж програміста. В середині 50-тих років, група програмістів під керівництвом Джона Бакуса в ІВМ висунули ідею, яка повністю змінила бачення та природу комп'ютерів. Бакус та його колеги думали, а чи можна написати чисто математичний код та формули і нехай сам комп'ютер перекладає його у

машинний код. В 1955 команда створила першу версію Smalltalk, який був першим прикладом високорівневої мови програмування. З того часу багато мов було створено, більшість з яких перевершували минулі. C++ також була в цій гілці еволюції. Одним з її предків була мова C, яка була створена Bell Laboratories Денісом Рітше в 1972 році, а далі стандартизована American National Standard Institute в 1989 [19].

Не зважаючи на багато привабливих речей і високу інтерактивність середовища користувача, процес програмування Smalltalk ніколи не був комерційно успішним. Професіонали проявляли свій інтерес до об'єктно-орієнтованого програмування тільки після того, як центральні ідеї були імплементовані в варіації C, які вже стали стандартом індустрії. Але було кілька паралельних спроб створити об'єктно-орієнтовану мову на базі C. Найбільш успішним став C++, який створив Бйорне Струстрап в Bell Laboratories в ранні 80-ті. C++ включав в себе стандарти з мови C, які давали змогу легко інтегрувати C++ код в уже існуючі C програми [20].

Коли ви пишете програму на C++, ваш перший крок це створення файлу, який має містити текст програми, який називається source файл. Після того ви можете запустити вашу програму, в якій проходить процес перекладу source файлу в скомпільовану форму. Перший крок в цьому процесі – запустити програму, яка називається компілятор, яка у свою чергу перекладає source файл в object файл, який містить у собі машинний код та інструкції. Цей файл потім комбінується з іншими файлами щоб створити відтворюваний файл, який може бути запущений в системі. Інші object файли, що створені раніше, називаються бібліотеками, які мають машинний код для різних операцій, які потрібні програмі. Цей процес комбінації всіх індивідуальних об'єктів називається поєднання [24].

Платформа на базі Arduino.

Проект Arduino – це компанія з відкритим кодом апаратного та програмного забезпечення, проекти та спільнота користувачів, яка розробляє та виробляє

одноплата мікроконтролери та комплекти мікроконтролерів для створення цифрових пристроїв. Його апаратні продукти ліцензуються згідно з ліцензією CC BY-SA, тоді як програмне забезпечення ліцензується відповідно до GNU Lesser General Public License (LGPL) або GNU General Public License (GPL), що дозволяє будь-кому виготовляти плати Arduino та розповсюджувати програмне забезпечення. Плати Arduino доступні на офіційному веб-сайті або через авторизованих дистриб'юторів [21].

Конструкції плат Arduino використовують різні мікропроцесори та контролери. Плати оснащені наборами контактів цифрового та аналогового входу/виводу (I/O), які можна підключати до різних плат розширення («щитів») або макетних плат (для створення прототипів) та інших схем. Плати мають інтерфейси послідовного зв'язку, включаючи універсальну послідовну шину (USB) на деяких моделях, які також використовуються для завантаження програм. Мікроконтролери можна програмувати за допомогою мов програмування C і C++, використовуючи стандартний API, який також відомий як мова Arduino Ino. На додаток до використання традиційних інструментів компілятора, проєкт Arduino надає інтегроване середовище розробки (IDE) і інструмент командного рядка, розроблений у Go.

Проєкт Arduino розпочався в 2005 році як інструмент для студентів Інституту проєктування взаємодії Івреа, Італія, з метою забезпечити недорогий і простий спосіб для новачків і професіоналів створювати пристрої, які взаємодіють з навколишнім середовищем за допомогою датчиків і приводів. Типові приклади таких пристроїв, призначених для любителів-початківців, включають прості роботи, термостати та детектори руху.

Середовище розробки Arduino IDE.

Середовище розробки Arduino складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виведення тексту (консолі), панелі інструментів з кнопками команд, що часто використовуються, і кількох меню. Для завантаження програм та зв'язку середовище розробки

підключається до апаратної частини Arduino [25]. Програма, написана серед Arduino, називається скетч. Скетч пишеться в текстовому редакторі, який має інструменти вирізки/вставки, пошуку/заміни тексту. Під час збереження та експорту проекту в області повідомлень з'являються пояснення, а також можуть відображатися помилки. Вікно виведення тексту (консоль) показує повідомлення Arduino, які містять повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевірити та записати програму, створити, відкрити та зберегти скетч, відкрити моніторинг послідовної шини.

Бібліотеки додають додаткову функціональність скетчам, наприклад, при роботі з апаратною частиною або з обробкою даних.

Було вибрано середовище розробки Arduino IDE як безкоштовне середовище розробки для плат Arduino, яке включає редактор коду, компілятор, програматор та відлагоджувач. У ньому є весь необхідний мінімум для розробки програм: написання коду, перевірка коду, компіляція, завантаження скетчу, монітор послідовного порту.

У процесі написання коду проекту було вирішено відмовитися від сторонніх бібліотек для роботи з двигунами на користь написання свого класу Motor.

Переваги написання свого класу.

1. Створюються лише необхідні функції. Це зменшує складність проекту та робить код простіше для розуміння. Враховуються особливості обладнання (конкретні драйвери, нестандартні схеми підключення).

2. Свій клас зазвичай займає менше пам'яті та працює швидше, тому що в ньому немає зайвого функціоналу, який часто буває у сторонніх бібліотеках. Це особливо важливо для мікроконтролерів з обмеженими ресурсами (наприклад, ATmega328P в Arduino Uno).

3. Код можна легко налагодити і розширити. Якщо виникає помилка, то легко знайти причину.

4. Використання сторонніх бібліотек додає залежності, які можуть бути несумісними з іншими бібліотеками або конфліктувати з проектом.

5. Навчальна цінність створення свого класу можна глибше зрозуміти, як працює обладнання (наприклад, драйвер L298N). Це покращує навички програмування.

6. Можна легко адаптувати свій клас під нове обладнання або змінити логіку роботи, не залежно від оновлень сторонніх бібліотек.

Якщо необхідно швидко розробити робоче рішення, використання готової бібліотеки прискорить процес. У нашому проекті написання свого класу для роботи з DC-моторами на L298N – логічний варіант, тому що функціонал простий, а власний код дає більше контролю та легкість розуміння.

3.2 Розробка загального алгоритму

Для початку роботи з роботом його слід увімкнути. Відповідні роз'єми підключаються до акумуляторів та до контролера.

Керування мобільним роботом здійснюється за допомогою смартфона, на який встановлений додаток Arduino Bluetooth Controller (рис. 3.1).



Рисунок 3.1 – Зовнішній вигляд додатка Arduino Bluetooth Controller

3.2.1 Ініціалізація (setup)

Налаштування компонентів: дисплея, моторів, сервоприводів, Bluetooth та світлодіодної стрічки.

Виведення вітального повідомлення на екран.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

#include <Wire.h>
#include <Servo.h>
Servo up_down, left_right, front_back, clamp; // Создайте сервоприводный
объект
int clmp = 45;
int frnt_bck = 45;
int lft_right = 90;
int up_dwn = 45;

#include <SoftwareSerial.h> // для роботи з Bluetooth на довільних пінах
SoftwareSerial bluetooth(1, 0); // RX, TX
char command = ' '; // Команда з Bluetooth

#include <DHT11.h>
DHT11 dht11(12);
int temperature = 0;
int humidity = 0;

#include <FastLED.h>
#define LED_PIN 13 //пін, до якого підключена стрічка
#define LED_COUNT 12
CRGB leds[LED_COUNT];
int currentLED = 0; // Змінна для поточного світлодіода

previousMillis_rgb = 0; // останній час оновлення
const long interval_rgb = 30; // Інтервал у мілісекундах (100 мс)
unsigned long previousMillis_lcd = 0;
const long interval_lcd = 200;

const int leftSideSensor = A3; // Лівий бічний датчик
const int leftFrontSensor = A2; // Передній лівий датчик
const int rightFrontSensor = A1; // Передній правий датчик
const int rightSideSensor = A0; // Правий бічний датчик
const int threshold = 500; // Порогове значення для датчиків
```

3.2.2. Основний цикл (loop)

Оновлення RGB світлодіодів: світлодіоди перемикаються на заданий інтервал часу, створюючи ефект руху світла.

Оновлення даних із датчика температури та вологості: значення температури та вологості відображаються на дисплеї.

```

lcd.init();
lcd.backlight(); // Увімкніть підсвічування дисплея
lcd.print("LCD працює");

FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, LED_COUNT); //
ініціалізація стрічки
FastLED.clear(); // Гасимо всі світлодіоди
FastLED.show();

bluetooth.begin(9600); // ініціалізація Bluetooth
}

void loop() {
unsigned long currentMillis = millis(); // таймер реберної стрічки

if (currentMillis - previousMillis_rgb >= interval_rgb) { //Перевірте, чи ми
передаємо час для перемикавання
previousMillis_rgb = currentMillis; // оновлення часу
leds[currentLED] = CRGB::Black; // гасити попередній світлодіод
leds[6 + currentLED] = CRGB::Black;
currentLED++; // перехід до наступного світлодіода
if (currentLED >= LED_COUNT) {
currentLED = 0; // Якщо ми дійшли до кінця, ми повертаємось до початку
}
leds[currentLED] = CRGB(255, 40, 0); // Запалюємо світлодіодного
помаранчевого кольору
leds[6 + currentLED] = CRGB(255, 50, 50); // малиновий колір (RGB)
FastLED.show();
}

if (currentMillis - previousMillis_lcd >= interval_lcd) { //таймер для обробки
температури датчика вологи
previousMillis_lcd = currentMillis;

int result = dht11.readTemperatureHumidity(temperature, humidity);
if (result == 0) {
lcd.clear();
lcd.setCursor(1, 0);
}
}
}

```

```

    String s = String("Тепло :")+temperature+ String(" °C");
    lcd.print(s);
    lcd.setCursor(1, 1);

    s = String("Волога:")+humidity+ String(" % ");
    lcd.print(s);
}

}
// Значення читання з ІЧ -датчиками
int leftFrontValue = analogRead(leftFrontSensor);
int rightFrontValue = analogRead(rightFrontSensor);
int leftSideValue = analogRead(leftSideSensor);
int rightSideValue = analogRead(rightSideSensor);

// Автономний рух (обхід перешкод)
if (cmode == 3) {
    if (leftFrontValue < threshold && rightFrontValue < threshold) {
        turnAround();
    } else if (leftFrontValue < threshold) {
        turnRight();
    } else if (rightFrontValue < threshold) {
        turnLeft();
    } else if (leftSideValue < threshold) {
        moveRight();
    } else if (rightSideValue < threshold) {
        moveLeft();
    } else {
        moveForward();
    }
}
}

// Функції керування моторами
void moveForward() {
    leftMotor.setSpeed(150);
    rightMotor.setSpeed(150);
}
void moveBackward() {
    leftMotor.setSpeed(-150);
    rightMotor.setSpeed(-150);
}
void turnLeft() {

```

```

    leftMotor.setSpeed(-100);
    rightMotor.setSpeed(100);
}
void turnRight() {
    leftMotor.setSpeed(100);
    rightMotor.setSpeed(-100);
}
void turnAround() {
    leftMotor.setSpeed(-150);
    rightMotor.setSpeed(-150);
    delay(1000);
    leftMotor.setSpeed(-100);
    rightMotor.setSpeed(100);
    delay(1000); // Час повороту
}
void moveLeft() {
    leftMotor.setSpeed(50);
    rightMotor.setSpeed(150);
}
void moveRight() {
    leftMotor.setSpeed(150);
    rightMotor.setSpeed(50);
}
void stopMotors() {
    leftMotor.setSpeed(0);
    rightMotor.setSpeed(0);
}
//Обробка команд Bluetooth для візка
void TruckCommand(char cmd) {
    switch (cmd) {
        case 'F':moveForward(); break; // Вперед
        case 'B': moveBackward();break; // Назад
        case 'L':turnLeft();break; // повернути праворуч
        case 'R':turnRight();break; // Поворот направо
        default:stopMotors();break;
    }
}
// Обробка команд Bluetooth для манипулятора
void ManipulatorCommand(char cmd) {
    switch (cmd) {
        case 'F': frnt_bck += 3; break; // Вперед
        case 'B': frnt_bck -= 3; break; // Назад
        case 'L': lft_rght += 3; break; // повернути ліворуч
        case 'R': lft_rght -= 3; break; // повернути праворуч
    }
}

```

```

case 'U': up_dwn += 3; break; // вгору
case 'D': up_dwn -= 3; break; // вниз
case 'O': clmp++; break; // Відкрити
case 'C': clmp--; break; // Закрити

```

Повністю скетч програми наведений у Додатку А.

На основі раніше описаних функцій були складені блок схеми алгоритму роботи модульного реконфігурованого робота. Логіка роботи модульного реконфігурованого робота представлена на рисунках 3.2-3.4.

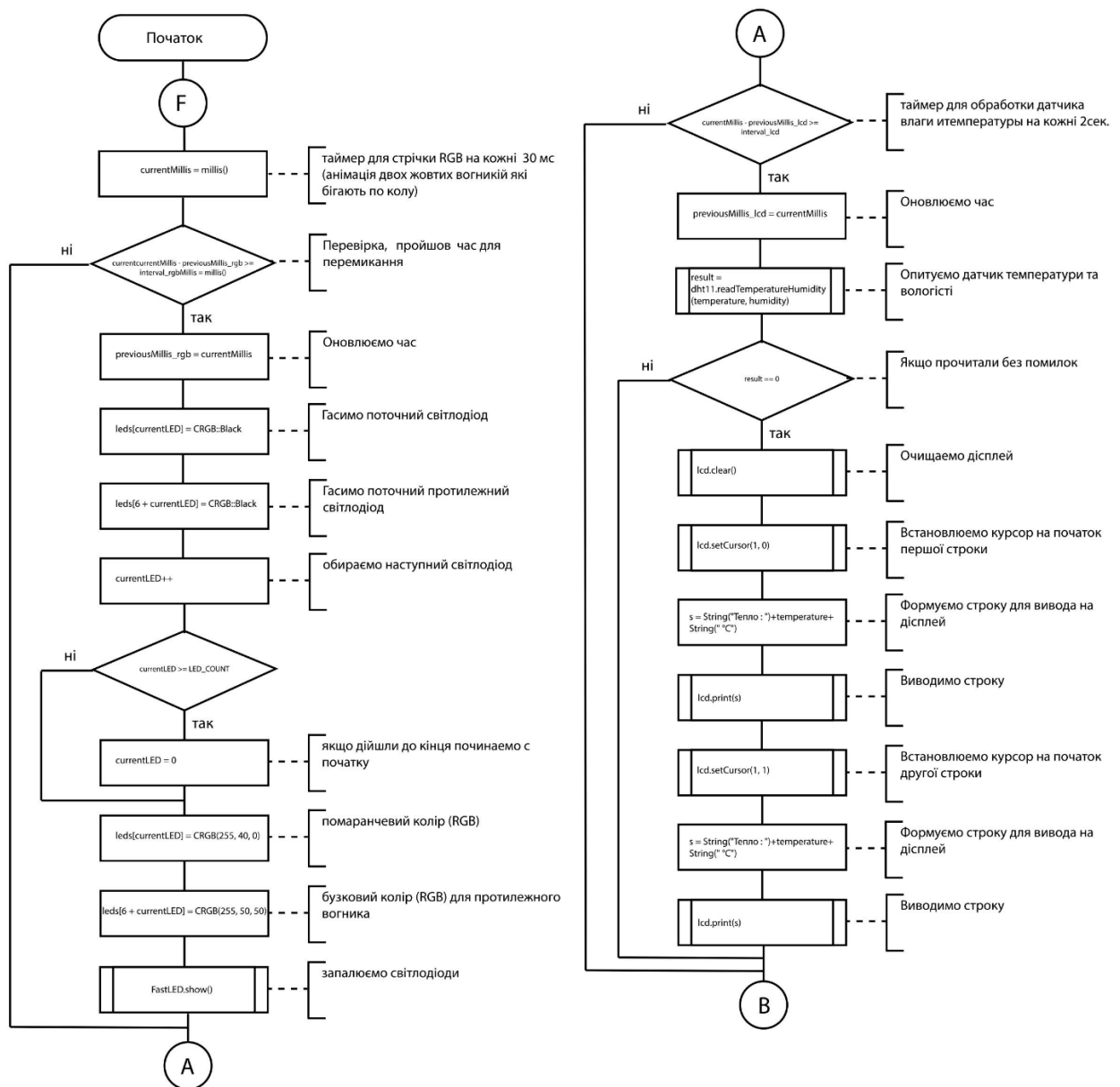


Рисунок 3.2 – Алгоритм роботи модульного реконфігурованого робота

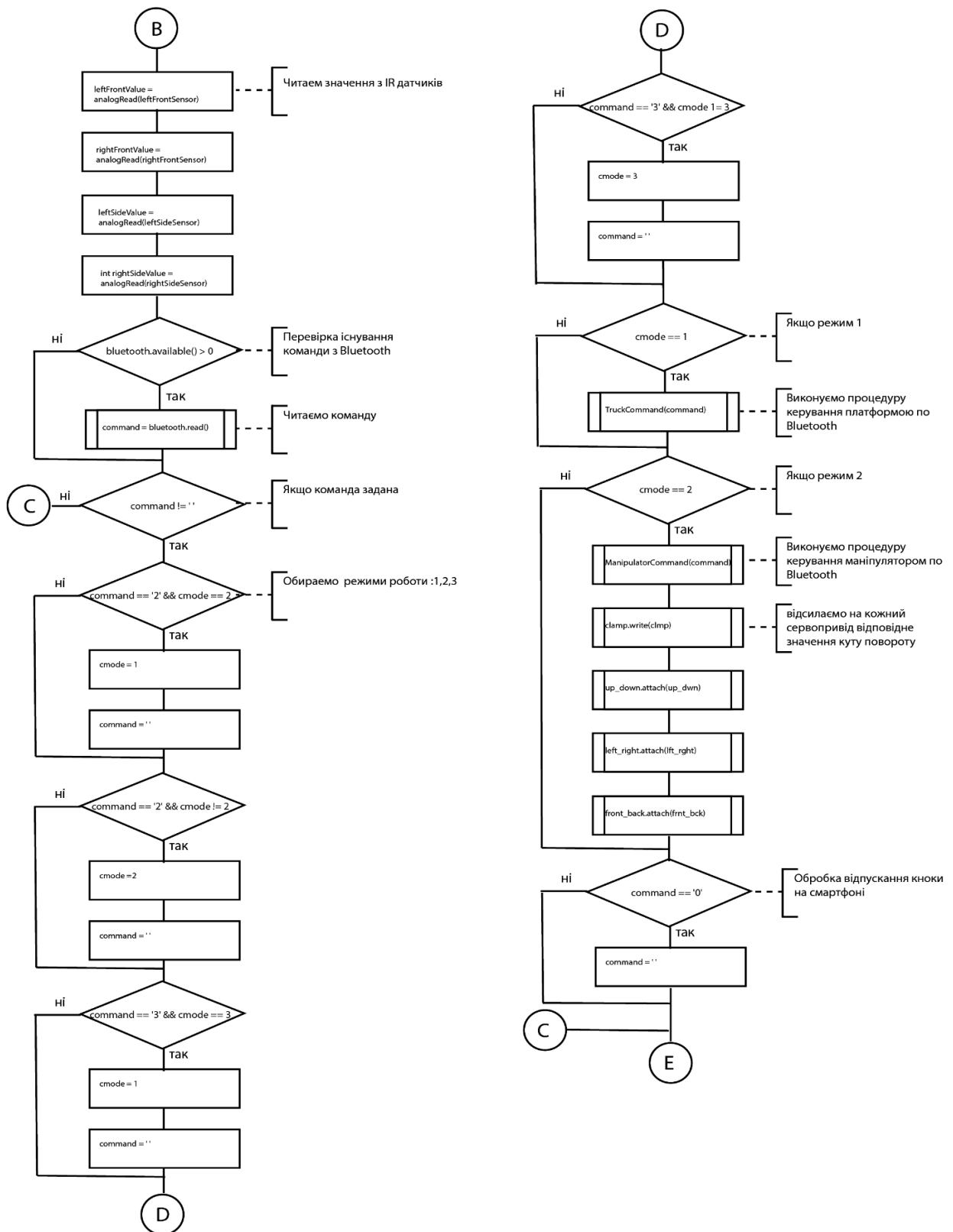


Рисунок 3.3 – Алгоритм роботи модульного реконфігурованого робота

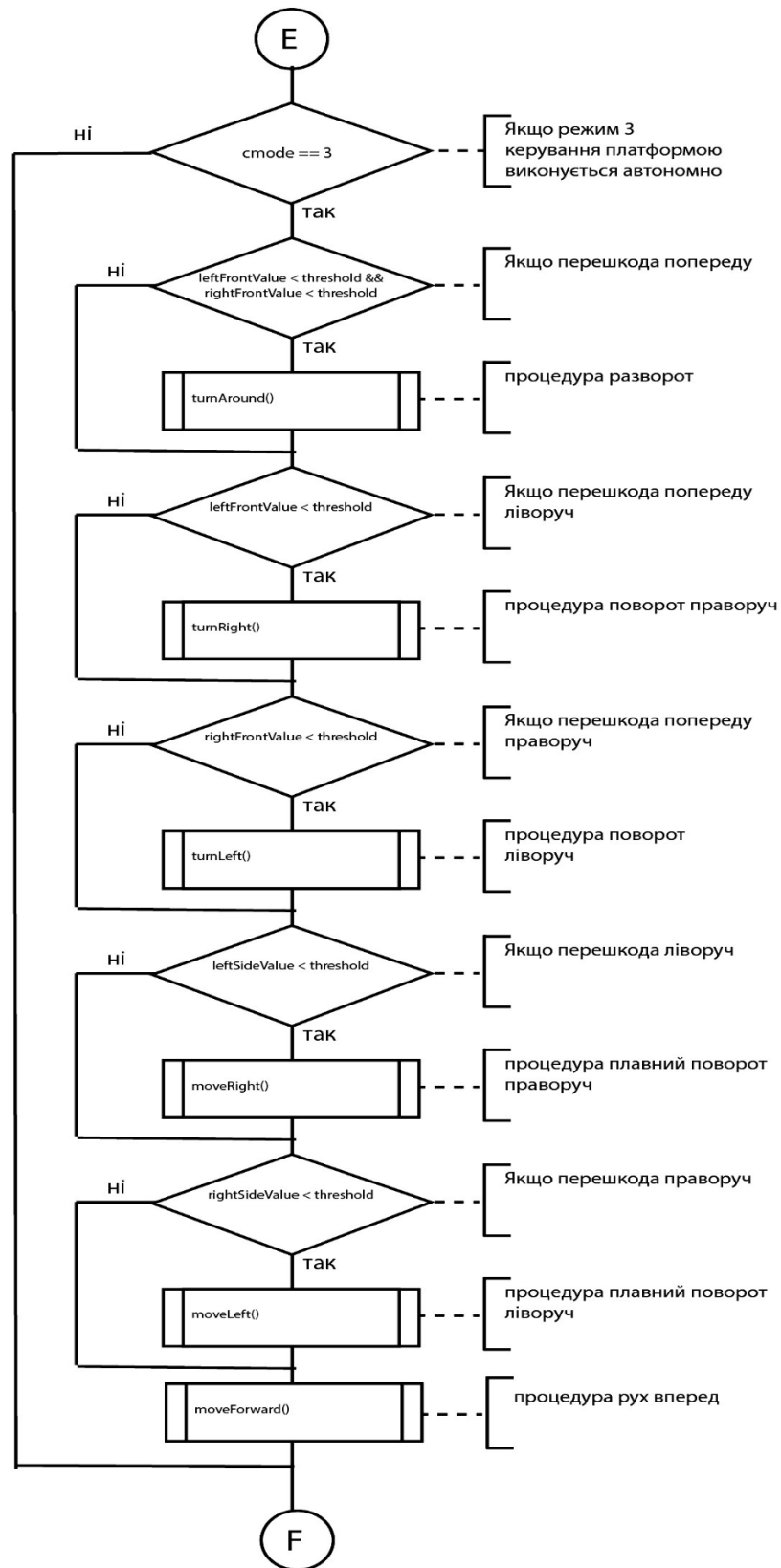


Рисунок 3.4 – Алгоритм роботи модульного реконфігурованого робота

3.2.3 Функції управління модульного реконфігурованого робота

Функції управління модульного реконфігурованого робота представлені на рисунках 3.5, 3.6.

Рух робота: реалізовані функції для руху вперед, назад, поворотів та зупинки.

Обхід перешкод: логіка роботи з ІЧ-датчиками дозволяє роботу визначати перешкоди та змінювати напрямок руху.

Маніпулятор: керування положенням маніпулятора та захопленням за допомогою команд Bluetooth.

Змінні та налаштування модульного робота:

- швидкість двигунів: керується через функцію `setSpeed()`;
- порогові значення датчиків: задані константою `threshold = 500`;
- таймери: використовуються для циклічного оновлення RGB-стрічки та дисплея.

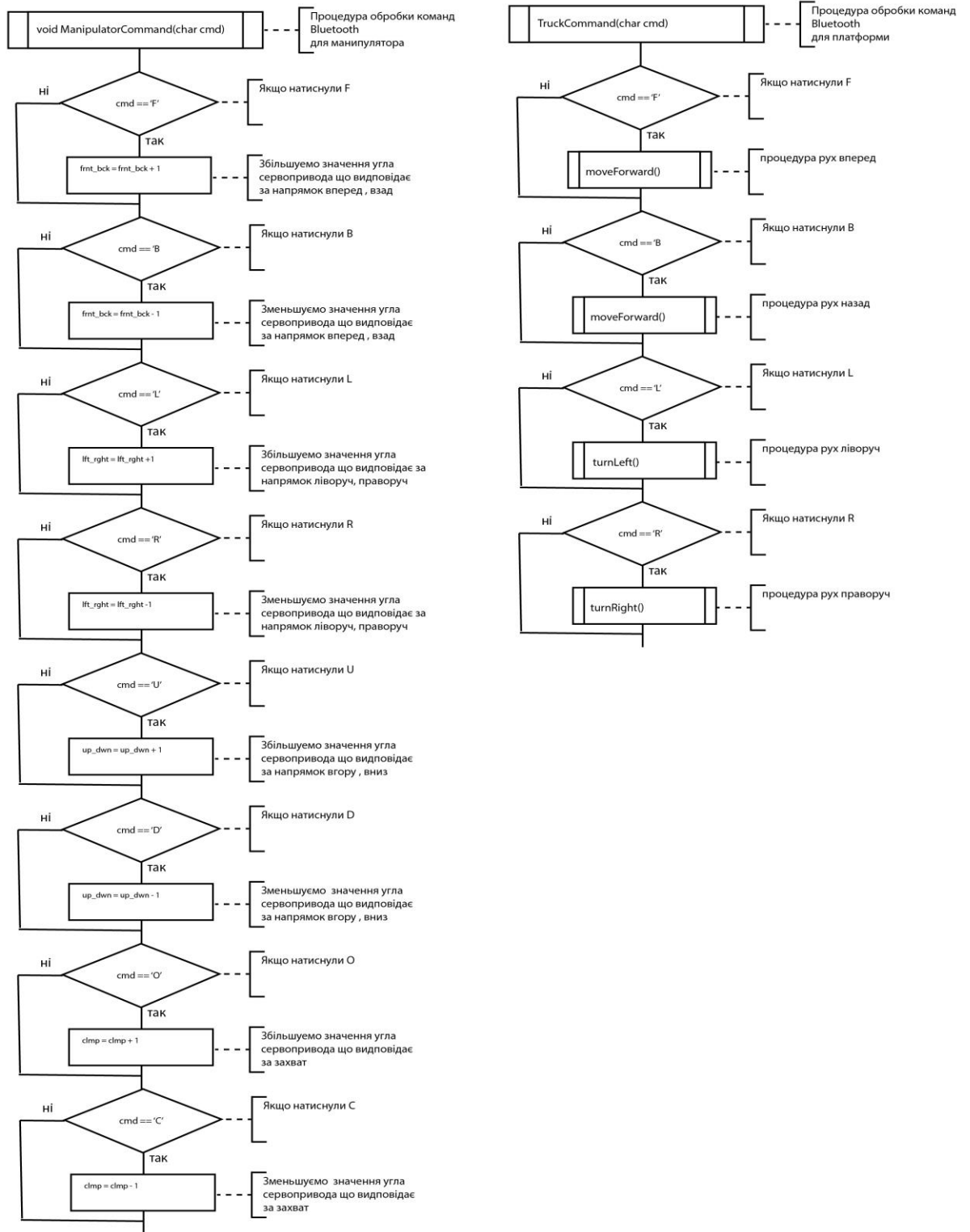


Рисунок 3.5 – Функції роботи модульного реконфігурованого робота

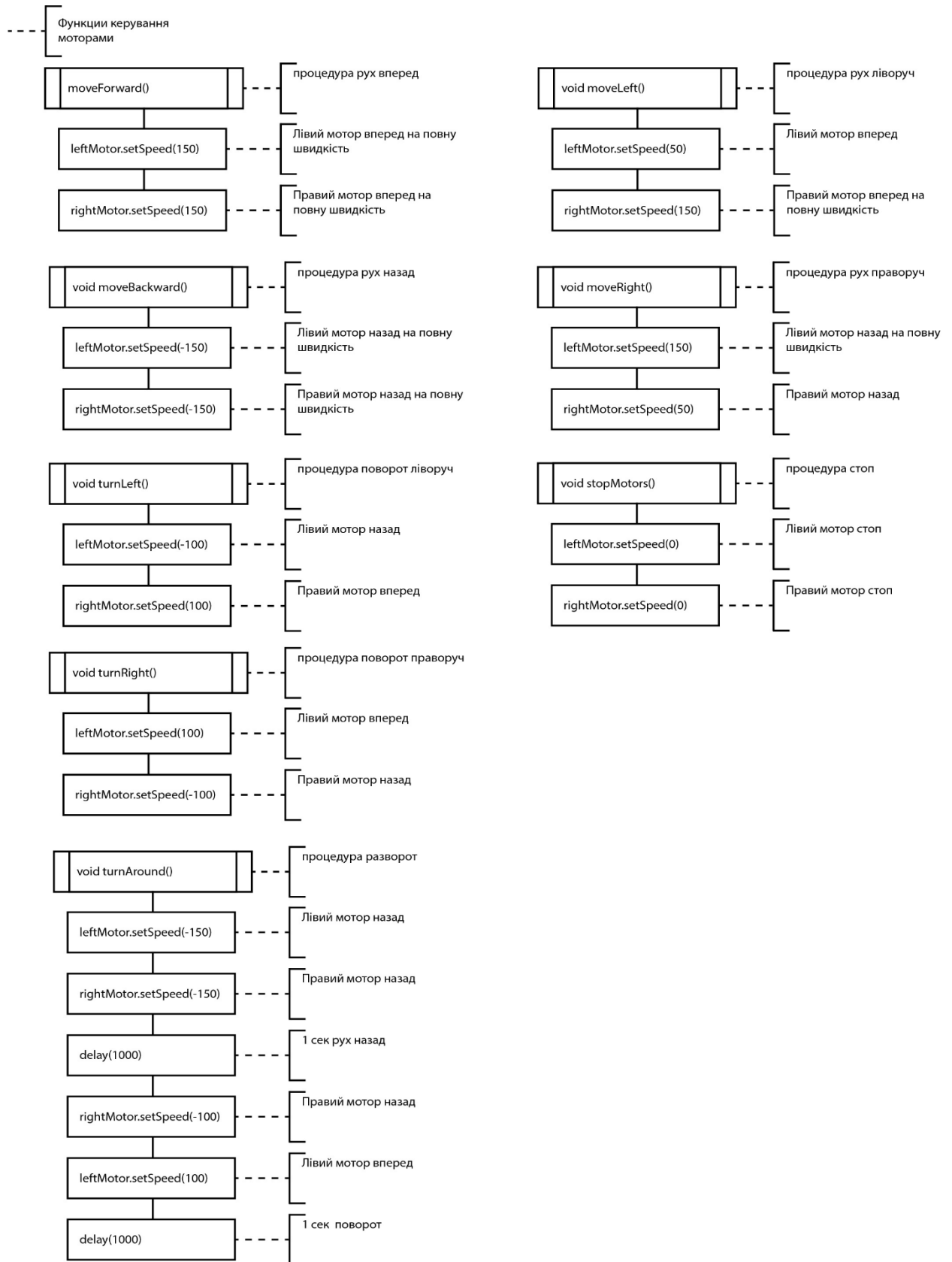


Рисунок 3.6 – Функції роботи модульного реконфігурованого робота

3.3 Приклад програмної реалізації функцій String та часу

Функція String.

Рядки – це текст, укладений у подвійні лапки: "Hello, World!". У нас є два набори інструментів для роботи з ними:

- статичні рядки – вони ж масиви символів char, є стандартними для мови C/C++ та працюють однаково на будь-якій платформі;

- динамічні String – рядки, в Arduino за них відповідає окрема бібліотека, яка входить до складу "ядра". Ці рядки прості та зручні у використанні, тому спочатку розберемо роботу з ними.

Створення String: рядок створюється як звичайна змінна, при створенні також можна надати рядку значення. String дозволяє автоматично перетворювати будь-який стандартний тип даних у рядок:

```
String s1 = 12345;
// із цілого числа
String s2 = 3.14; // з float
String s3 = "text"; // з рядкової константи
String s4 = F("text"); // з рядкової константи з F
String s5 = 's'; // із символу
int val = 1234;
String s6 = val; // зі змінної
String s7 = s6; // з іншого рядка
```

До рядка можна додати будь-який тип даних, так само як при створенні, по одній дії у рядку коду:

```
String s;
s += 12345; // із цілого числа
s += 3.14; // з float
s += "text"; // з рядкової константи
s += F("text"); // з рядкової константи з F
```

```
s += 's'; // із символу
int val = 1234;
s += val; // Зі змінної
String s2(1234);
s += s2; // з іншого рядка
```

Також String дозволяє складати рядки між собою за допомогою оператора +=. Дані мають будь-який тип, з яким String підтримує додавання.

Одним із доданків має бути рядок, як вищий:

стрінг + дані або дані + стрінг. Операція додавання повертає рядок назад, що дозволяє зробити "каскад" з таких додавань і зібрати рядок "одним рядком коду", складання відбувається зліва направо: стрінг + дані1 + дані2 або дані1 + стрінг + дані2 + дані3

Результат усієї суми можна:

– прирівняти до String:

стрінг = стрінг + дані1 + дані2

– надіслати у функцію, яка приймає String:

f(стрінг + дані1 + дані2)

```
String s;
s = String("Hello") + ',' + "World" + 1234 + 3.14;
Serial.println(s + "text" + 1234);
```

Доступ до символів. До рядка можна звернутися як до масиву та прочитати або змінити символ по порядку:

```
String s("hello!");
s[0] = 'H';
// Тут s == Hello!
char c = s [1];
// тут c == e
```

Порівняння String. Стрінги можна порівнювати між собою та зі звичайними рядками (const char*):

```
String s1 = "text1";
String s2 = "text2";
if (s1 == s2); // false
if (s1 == "text1"); // true
if (s2! = "text2"); // false
```

Проблеми та оптимізація String – перевага стрінгів полягає в тому, що з ними дуже легко та зручно працювати: збирати з інших рядків та змінних будь-яких типів, складати між собою, ділити на підрядки тощо. За зручність доводиться платити: String є динамічним об'єктом, що спричиняє деякі проблеми. Розглянемо всі разом.

String – важкий. Безперечно – використання String-рядків відразу додає пару кілобайт Flash пам'яті до ваги програми, оскільки для роботи з ними використовується менеджер пам'яті (вбудована бібліотека). У той же час, якщо програма вже використовується, динамічне виділення пам'яті – додавання String буде проблемно. На цьому дана проблема закінчується, оскільки якщо відкрити реалізацію бібліотеки String, можна побачити, що ці дії з рядками виконуються за допомогою стандартних рядкових функцій мови C.

String – повільний. Так, коли рядок змінює свою довжину – він починає змінювати свій розмір і навіть місце в оперативній пам'яті мікроконтролера. Переписування і перерозподіл пам'яті відбувається не миттєво, тому операції з String виконуються відносно довго: сотні мікросекунд. Якщо збирати рядок посимвольно – кожна добавка буде виконуватися довше, чим хотілося б.

Цього можна уникнути, використовуючи метод `reserve()`, який зарезервує пам'ять для збільшення рядка без виділення пам'яті. Якщо місце під рядок зарезервовано – операції із рядком виконуватимуться з такою ж швидкістю, як і зі звичайними рядками, тому що для них використовуються самі стандартні рядкові функції.

String – небезпечний. Неакуратна робота зі String може призвести до сильної фрагментації пам'яті, неправильної роботи програми та навіть повного її

зависання. У той же час, якщо розуміти як працюють стрінги та використовувати ефективні та безпечні конструкції в роботі з ними – можна уникнути абсолютно всіх проблем.

Найпростішою з точки зору використання функцією часу є затримка: програма "зависає" всередині функції затримки і чекає на вказаний час. Затримка дозволяє дуже зручно та наочно організувати роботу простої "однозадачної" програми. Є два варіанти затримок:

```
delay(time)
void setup() {}
void loop() {
  // що то виконати
  delay(500); // зачекати півсекунди
}
void yield() {
  // ваш код
}
uint32_t sec = millis() / 1000ul; // повна кількість секунд
int timeHours = (sec / 3600ul); // годинник
int timeMins = (sec % 3600ul) / 60ul; // хвилини
int timeSecs = (sec % 3600ul) % 60ul; // секунди
```

Затримка на вказану кількість мілісекунд (мс). 1 секунда = 1000 мс.

time приймає тип даних unsigned long і може призупинити виконання терміном від 1 до 4 294 967 295 мс (~50 діб) з роздільною здатністю 1 мс. Працює на системному таймері, тому не працює всередині переривання та при відключених перериваннях. delayMicroseconds(time) затримка на вказану кількість мікросекунд (мкс). 1 секунда = 1000000 мкс.

time приймає тип даних unsigned int і може призупинити виконання терміном від 4 до 16383 мкс (так, менше ніж максимум для цього типу даних) з роздільною здатністю 4 мкс.

Працює не на таймері, а на пропуску тактів процесора, тому може працювати у перериванні та при відключених перериваннях. Іноді не зовсім коректно працює зі змінними, тому потрібно намагатися використовувати константи (const чи просто число).

Часто використовують у бібліотеках для емуляції цифрових інтерфейсів зв'язку. Функція yield() Розробники Arduino подбали про те, щоб функція delay() не просто блокувала виконання коду, але й дозволяла виконувати інший код під час цієї затримки. Ця функція отримала назву yield() і працює наступним чином: то розташований в ній код буде виконуватися під час роботи будь-якої затримки delay() в програмі! Це рішення хоч і здається безглуздим, але в той же час дозволяє швидко і без написання зайвих таймерів реалізувати пару завдань, що паралельно виконуються. Це цілком відповідає ідеології Arduino – максимально проста та швидка розробка прототипу. Розглянемо простий приклад: стандартний миготливий світлодіод, але з опитуванням кнопки:

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, 1);
  delay(1000);
  digitalWrite(13, 0);
  delay(1000);
}

void yield() {
  // а тут можна опитувати кнопку
  // і не пропустити натискання з-за delay! }
```

Функції рахунку: дані функції повертають час, що минув з моменту запуску програми, так званий аптайм (англ. uptime). Таких функцій у нас дві:

– мілісекунди, тип `unsigned long`, від 1 до 4 294 967 295 мс (~50 діб), роздільна здатність 1 мс. Після "переповнення" відлік починається з нуля;

– мікросекунди, тип `unsigned long`, від 4 до 4 294 967 295 мкс (~70 хвилин), роздільна здатність 4 мкс.

Після "переповнення" відлік починається з нуля.

Обидві ці функції працюють у "фоні" на перериваннях таймера. Продовжують працювати всередині затримок `delay ()` і `delay Microseconds ()`, тобто, хід часу не збивається. Припиняють рахувати час, доки відключені переривання (через `noInterrupts()` чи `cli()`). Не змінюють свого значення всередині обробника переривання (зовнішнього, таймера), тому що переривання самі відключаються на час обробки. Можна викликати всередині обробника переривання. Ці функції дозволяють організувати програму практично будь-якої складності з будь-якою кількістю завдань, що паралельно виконуються за таймером.

Мілісекунди – не найзручніший спосіб оцінити час роботи програми. Можна перевести його в більш людський годинник – хвилини та секунди за допомогою нехитрих математичних операцій.

3.4 Висновки до третього розділу

У розділі була проведена розробка програмного забезпечення керування мобільним реконфігурованим роботом. Було вибрано середовище розробки – `Arduino IDE`, розроблені алгоритми роботи всіх систем реконфігурованого мобільного робота. У процесі написання коду проекту замість стандартних бібліотек було вирішено написати свій клас `Motor` для роботи з двигунами, що спростило функціонал, а власний код дав більше контролю та легкість розуміння.

Розроблений приклад програмної реалізації функцій String та часу. Ці функції дозволяють організувати програму практично будь-якої складності з будь-якою кількістю завдань, що паралельно виконуються за таймером.

Програма керування розділена на окремі логічні блоки (керування моторами, маніпулятором, датчиками, RGB-стрічкою), що спрощує налагодження та модернізацію робота. Використані стандартні бібліотеки (GyverMotor, Servo, FastLED), які підвищують надійність та гнучкість. Алгоритм реалізує три режими: ручне керування, керування маніпулятором та автономний рух. Це робить систему універсальною для різних сценаріїв. Bluetooth-модуль дозволяє керувати роботом на відстані, що є зручним для практичного застосування. Використання датчиків температури, вологості та ІЧ-сенсорів робить пристрій більш інтерактивним та здатним реагувати на навколишнє середовище. LCD-дисплей відображає поточні дані (температуру, вологість), а RGB-стрічка забезпечує візуальні ефекти, що покращує користувацький досвід. В автономному режимі робот здатний оминати перешкоди, використовуючи просту, але ефективну логіку на основі даних із ІЧ-датчиків. Легко змінити або розширити функціонал завдяки добре структурованому коду (наприклад, додати нові команди для Bluetooth або додаткові сенсори).

Можливість тонкого налаштування параметрів, таких як швидкість двигунів, порогові значення датчиків і команди Bluetooth, робить алгоритм придатним для різних застосувань.

Спроектований реконфігурований мобільний робот представляє додаткові можливості:

- додати обробку помилок для модулів Bluetooth та DHT11;
- розширити функціонал.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Експериментальне дослідження рухом мобільного робота

Практичного значення в різних областях техніки набуває задача управління рухом мобільних колісних роботів.

Мета експерименту – розробити та впровадити алгоритм управління рухом мобільного робота, здатного здійснювати автономне переміщення у просторі.

Алгоритм має забезпечувати:

- безпечне уникнення перешкод завдяки використанню даних, отриманих від інфрачервоних датчиків;
- оптимізацію маршруту руху для зменшення пройденої відстані та енергоспоживання.

Експеримент також спрямований на перевірку працездатності створеного алгоритму та системи в умовах, наближених до реальних, з використанням мобільного робота, побудованого на основі доступних компонентів і технологій.

Для обраного експерименту мобільний робот був зібраний з наступних компонентів (рис. 4.1):

- мікроконтролер Arduino Uno Rev3 (ATmega16U2);
- шилд підключення датчиків для Arduino Uno Rev3;
- акумулятор з зарядним модулем TP4056 Mini-USB з функцією захисту акумулятора;
- драйвер двигунів на базі чіпа L298;
- 4-канальний інфрачервоний модуль Arduino;
- робо-платформа 4-х колісна двопалубна повнопривідна.

Інфрачервоні датчики встановлені на мобільну платформу за схемою: два датчика на передньому боці платформи, два датчика з обох боків.



Рисунок 4.1 – Макет мобільного робота з інфрачервоним датчиком

Для того щоб математично описати рух мобільного робота треба визначити системи координат. Введемо дві системи координат – світову систему координат W (яка нерухома в просторі) і система координат робота R , яка переміщається в просторі і залишається нерухомою щодо самого робота (рис. 4.2). Необхідно визначити місце розташування робота, тобто перетворення координат між W та R . Якщо припустити, що робот обмежується переміщенням на поверхні, його місце розташування може бути визначене вектором стану, який складається з трьох параметрів:

$$X = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}, \quad (4.1)$$

де x і y визначають місце розташування зумовленою точки центру робота у світовій системі координат, θ визначає кут повороту між системами координат (кут між осями x^W та y^W).

Дві системи координат збігаються в момент, коли умовний центр робота знаходиться на початку координат та $x = y = \theta = 0$.

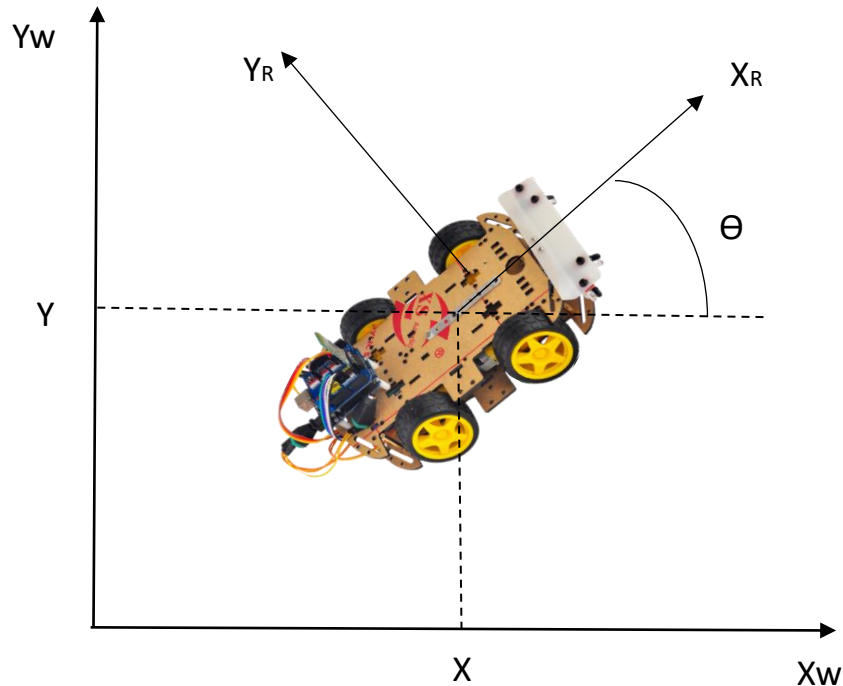


Рисунок 4.2 – Управління рухом при певному положенні мобільного робота

В інфрачервоному модулі встановлені інфрачервоні світлодіоди. Світло відображене від перешкоди, направляється на позиційно-чутливий фотоелемент. Від позиції падаючого на нього променя залежить його провідність. Провідність, в свою чергу, перетворюється в напругу і виводиться на аналоговий вихід.

Інфрачервоний датчик відстані має три контакти;

- живлення +5 В;
- земля;
- аналоговий вихід.

Аналоговий вихід датчика підключається до аналогового входу Arduino Uno.

Отриманий сигнал за допомогою аналогово-цифрового перетворювача (АЦП), що вбудовано в Arduino, перетворюється у напругу за наступною формулою:

$$V = \frac{k \cdot V_{\text{ref}}}{2^n}, \quad (4.2)$$

де V – розраховане значення вихідної напруги;

k – значення сигналу (цифровий код), отримане з АЦП;

V_{ref} – опорна напруга АЦП (зазвичай 5 В або 3,3 В залежно від налаштувань і моделі Arduino);

n – кількість бітів АЦП (наприклад, для Arduino Uno $n=10$, що означає 10-бітний АЦП, тобто $2^{10}=1024$ рівнів дискретизації).

Таким чином, формула для Arduino Uno, де $n=10$, спрощується до:

$$V = \frac{k \cdot V_{\text{ref}}}{1024}.$$

Наприклад, якщо $V_{\text{ref}}=5$ В і $k=512$, то:

$$V = \frac{512 \cdot 5}{1024} = 2,5 \text{ В}.$$

Вихідний сигнал, що отриманий з даного датчика, обернено пропорційний – зі збільшенням відстані його значення повільно зменшується.

Приклад графіку залежності напруги від відстані [25] наведено на рисунку 4.3.

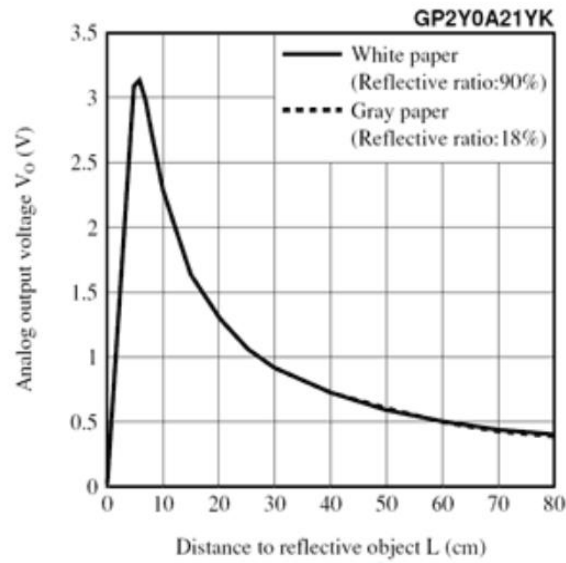


Рисунок 4.3 – Приклад залежності напруги від відстані [25]

Для отримання формули для розрахунку відстані в залежності від напруги проведено власний дослід, в якому знято залежність напруги від відстані.

Залежність зображена на рисунку 4.4.

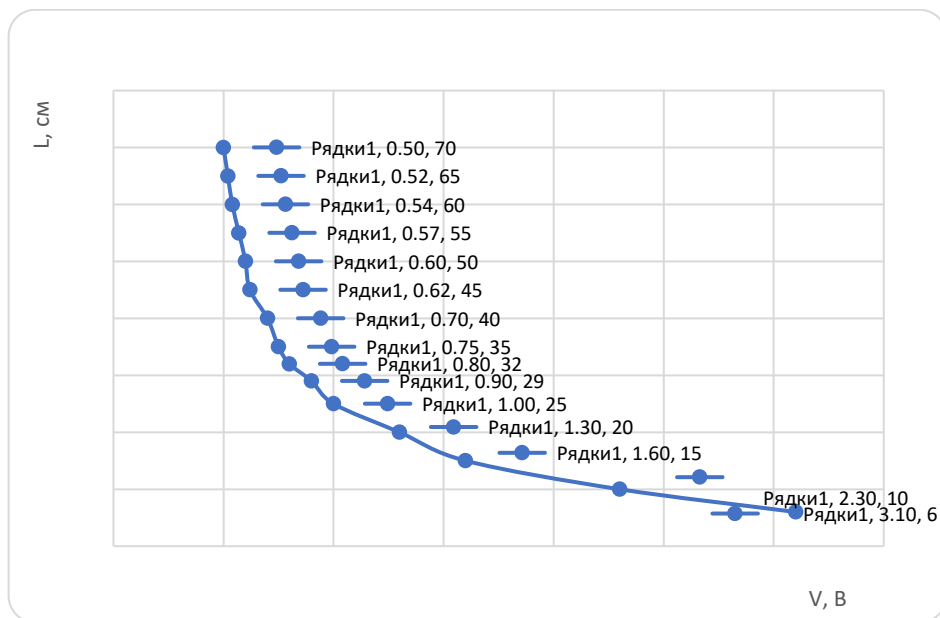


Рисунок 4.4 – Залежність напруги від відстані

Рух робота складається з прямолінійних відрізків окремо від розворотів на місці. Мобільний робот намагається звести до мінімуму загальну подолану

відстань, тому він завжди відразу повертається передньою частиною до наступної точки та їде прямо до неї.

На першому кроці управління руху припустимо, що поточний стан мобільного робота (x, y, Θ) і наступної точкою є (W_x, W_y) . Спочатку робот повинен повернутися до зазначеної точки. Вектор спрямування повинен вказувати на:

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} W_x - x \\ W_y - y \end{pmatrix}. \quad (4.3)$$

Абсолютне значення кута в градусах α , до якого мобільний робот повинен повернутися:

$$\alpha = \tan^{-1} \frac{d_y}{d_x}. \quad (4.4)$$

Необхідно переконатися, що α знаходиться в правильному квадранті $-\pi < \alpha \leq \pi/2$. Робот вже повернутий на певний кут, тому кут на який він повинен повернутися $\beta = \alpha - \theta$. Щоб робот рухався найбільш ефективно, потрібно змістити кут, додавши або віднімаючи 2π , щоб $-\pi < \beta \leq \pi$. Після цього, мобільний робот повинен рухатися по прямій на відстань:

$$d = \sqrt{d_x^2 + d_y^2}. \quad (4.5)$$

Тобто, алгоритм управління рухом мобільного робота полягає в тому, що робот сканує навколишнє середовище для визначення наявності перешкод на шляху (рис. 4.5).

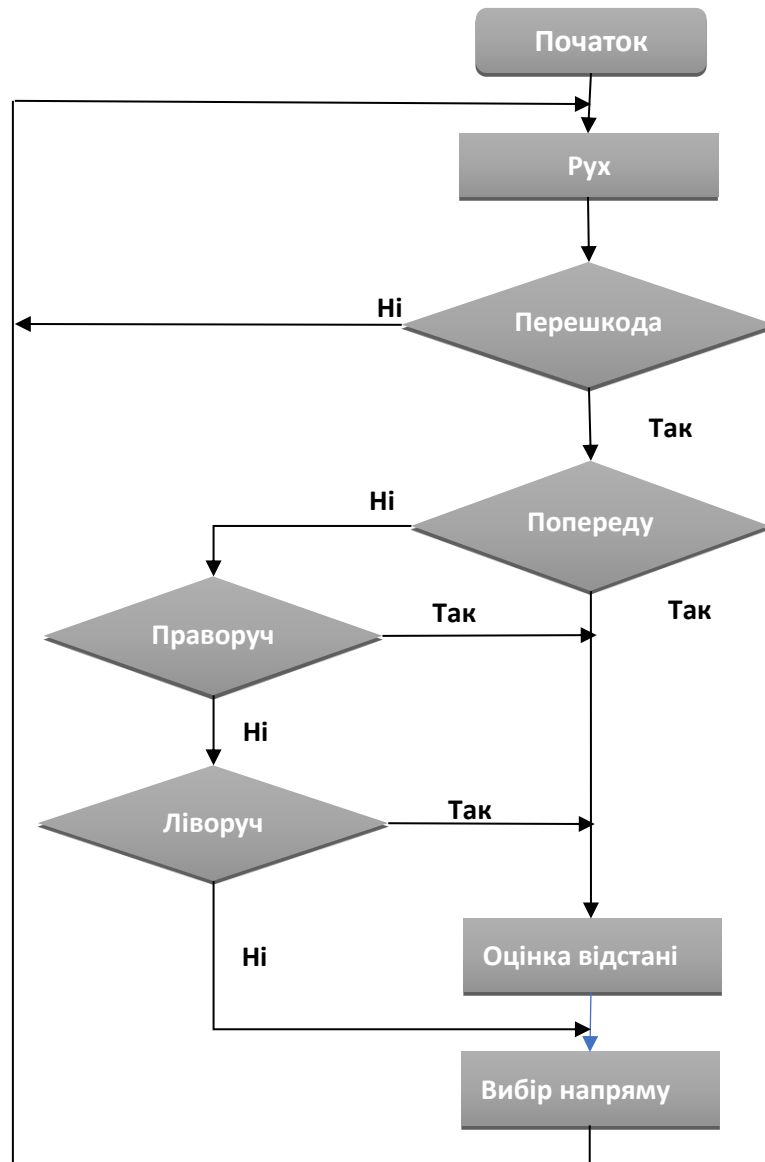


Рисунок 4.5 – Алгоритм управління рухом мобільного робота

Після знаходження шляху, на якому відсутні перешкоди, мобільний робот при русі електромоторів в різних напрямках робить розворот і починає рухатися вперед до тих пір, поки не зустрінеться з новою перешкодою. Згідно з розробленим алгоритмом було проведено тестування управління рухом мобільного робота, під час якого були взяті показання інфрачервоних датчиків руху мобільного робота.

Висновок. В результаті експерименту було розроблено алгоритм управління руху. Результати, отримані завдяки тестуванню розробленого

алгоритму, підтверджують прийнятність алгоритму, здатного автоматизувати пересування мобільного робота.

4.2 Аналіз експериментального дослідження

4.2.1 В результаті експерименту було розроблено алгоритм автономного управління рухом мобільного робота. Результати, отримані завдяки тестуванню розробленого алгоритму, підтверджують прийнятність алгоритму, здатного автоматизувати пересування мобільного робота.

У процесі експериментального дослідження проведений аналіз роботи датчиків відстані, розглянуто принципи їхньої дії, переваги, недоліки та області застосування в умовах навчальної робототехніки. У дослідженні акцент зроблено на лазерні, інфрачервоні та ультразвукові датчики, з урахуванням специфіки їх використання на навчальному роботі.

Особливості лазерних датчиків.

- висока точність, вузький кут огляду, менша схильність до перешкод;
- менш схильні до перешкод, але є дорожчими через складність технології.

Особливості інфрачервоних датчиків.

- обмежена точність для об'єктів, що пропускають або поглинають світло (скло, темні поверхні);
- чутливі до сонячного світла, що може викликати хибні спрацьовування;
- доступна ціна, компактність.

Ультразвукові датчики використовують звукові хвилі для вимірювання відстані до об'єкта або поверхні. Ці датчики ефективні для виявлення прозорих або відбиваючих поверхонь, а також об'єктів у середовищі з пилом і димом. Вони також мають високу чутливість до дрібних предметів і можуть виявляти їх на великій відстані. Вони зазвичай зустрічаються в таких додатках, як робототехніка, датчики паркування та системи уникнення перешкод.

Ультразвукові датчики також є перспективними, особливо для виявлення об'єктів на середніх дистанціях, але їхня залежність від зовнішніх факторів є критичним обмеженням.

Особливості ультразвукових датчиків:

- менша залежність від типу поверхні об'єкта;
- чутливість до змін параметрів навколишнього середовища (температура, вологість);
- обмежена дальність через розсіювання звуку.

З точки зору продуктивності, важко визначити, який датчик кращий – інфрачервоний чи ультразвуковий, оскільки обидва мають свої сильні та слабкі сторони. Однак, коли йдеться про складніші та різноманітні середовища, ультразвукові датчики, як правило, працюють краще, ніж ІЧ-датчики. Це пояснюється тим, що ультразвукові датчики покладаються на звукові хвилі, а не на відбите світло, тому на них менше впливають зміни умов освітлення та властивостей поверхні.

В умовах навчальної робототехніки найбільш раціональним є використання інфрачервоних датчиків, таких як F233-01, через їхню доступність, достатню точність і невисоку вартість. Вибір цього датчика зумовлений відповідністю вимогам:

- невелика робоча дистанція (до 80-100 см);
- компактність і простота інтеграції;
- здатність працювати в закритих приміщеннях.

На основі проведеного аналізу для використання в навчальному чотириколісному роботі обрано інфрачервоний датчик F233-01. Він забезпечує оптимальне співвідношення ціни, точності та функціональних можливостей для виконання завдань у невеликому закритому приміщенні. Застосування цього датчика дозволить реалізувати базові функції навігації та уникнення перешкод, що є ключовими для навчальних проєктів. Водночас, при модифікації алгоритмів управління роботом і використанні інших типів датчиків відстані,

таких як лазерні або ультразвукові, можна досягти більш інтелектуальної та адаптивної навігації. Наприклад, лазерні датчики забезпечують високу точність і вузький кут огляду, що ідеально підходить для складних маршрутів, а ультразвукові – дозволяють виявляти об'єкти на середніх дистанціях незалежно від типу поверхні. Такий підхід розширить функціональність робота та зробить його здатним вирішувати завдання в більш складних умовах.

4.2.2 Переваги виконаного проєкту у порівнянні з LEGO MINDSTORMS® EV3 та Makeblock mBot v1.1.

– LEGO MINDSTORMS® EV3 – готове рішення на своїх контролерах, є закритою платформою. Одна з властивостей закритих платформ - таких, як "Лего" - несумісність їх компонентів з іншими платформами. З'єднання, які використовують у «Лего», у реальному світі не зустріти, також ніде не використовується і мова програмування лего-роботів. Його можна використовувати лише як перший крок на шляху до робототехніки. На складання моделі LEGO MINDSTORMS® EV3, в середньому, потрібно біля 35 хвилин. Ціна комплекту 49000 грн.

– Makeblock mBot v1.1 – програмований робот-конструктор на Arduino Uno для використання у школі з додатковими модулями: Датчик освітлення ультразвуковий датчик, бездротове управління Bluetooth. Програма для Makeblock mBlock – графічна, тому учні зможуть легко засвоїти правила керування роботом. mBot може пересуватись по заданій траєкторії, помічати перешкоди і оминати їх. На складання моделі Makeblock mBot v1.1, в середньому, потрібно 25-30 хвилин. Ціна конструктора 9000 грн.

– Робототехнічна платформа на основі Arduino, яка використовується в нашому проєкті, значно дешевша порівняно з рішеннями на платформі LEGO Mindstorms або аналогами. Це дозволяє знизити фінансові витрати на впровадження робототехнічних курсів у навчальний процес. Arduino пропонує ширші можливості для налаштування і модифікації. У нашому проєкті передбачена можливість підключення різних типів датчиків (інфрачервоні,

лазерні, ультразвукові), що забезпечує гнучкість у виборі обладнання залежно від навчальних задач і умов використання. LEGO має обмежений набір модулів, що обмежує експерименти з обладнанням. Використання Arduino знайомить студентів із реальними принципами проектування електроніки, написанням коду на C/C++, підключенням модулів і датчиків. LEGO, навпаки, пропонує більш інтуїтивно зрозумілу та спрощену платформу, яка часто приховує технічні аспекти процесів.

Відмінність проекту від аналогічних платформ на основі Arduino полягає у використанні корпусу для електроніки з вентиляцією, до якого за допомогою конекторів jack 3,5mm підключаються всі модулі (рис. 4.6). Кожен модуль має індивідуальне гніздо, яке забарвлене відповідним кольором. Датчик температури та вологості, rgb-стрічка, LCD-дисплей, маніпулятор підключаються до гнізд на верхній частині корпусу, драйвер двигуна та інфрачервоні датчики - до гнізд на задній частині корпусу. Також на корпусі знаходяться два гнізда для підключення додаткових модулів. На складання моделі, в середньому, потрібно 20 хвилин.

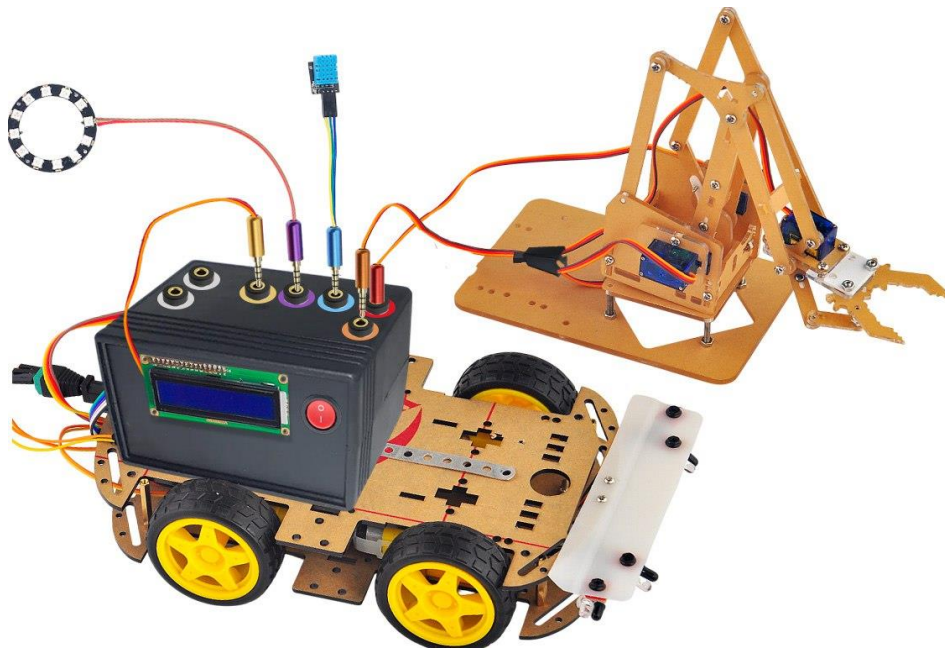


Рисунок 4.6 – Підключення модулів до корпусу для електроніки

Це значно зменшує час на реконфігурацію моделі мобільного робота. Таким чином, наш проєкт на базі Arduino є більш доступним, універсальним та орієнтованим на реальну інженерну підготовку, що забезпечує глибше залучення студентів до технічних аспектів робототехніки.

Наукова цінність проєкту полягає в розробці, дослідженні та впровадженні ефективного алгоритму управління рухом мобільного робота, який поєднує в собі базові принципи робототехніки та елементи інтелектуальної навігації.

Основними аспектами наукової складової є:

– розробка алгоритму управління рухом мобільного робота. Алгоритм дозволяє роботу адаптуватися до умов навколишнього середовища, використовуючи інфрачервоні датчики, які зчитують інформацію про відстань до перешкод, і приймати рішення про напрямок руху;

– реалізація алгоритму перемикання між режимами роботи однією кнопкою.

Універсальність коду полягає у створенні функціонального блоку, який дозволяє змінювати режими роботи мобільного робота за допомогою однієї кнопки. Це рішення є компактним і економічним, що робить його зручним для використання у навчальних проєктах. Перемикання між режимами, такими як автономна навігація, ручне управління або обхід складних маршрутів, дає можливість демонструвати різні стратегії поведінки робота;

– аналіз впливу типу датчиків на якість і точність навігації. У ході дослідження було проведено порівняння різних типів сенсорів (інфрачервоні, лазерні, ультразвукові) та їхньої ефективності в умовах закритого простору. Це дозволило визначити оптимальний вибір сенсорів для навчальних цілей;

– можливість масштабування і модифікації алгоритму. Розроблений алгоритм легко адаптується до використання інших типів датчиків або інтеграції додаткових функцій, що дозволяє досягти більш інтелектуальної навігації, такої як побудова оптимального маршруту або розпізнавання складних перешкод.

4.3 Розрахунок штучного освітлення робочого місця монтажу та експлуатації реконфігурованим мобільним роботом

Ціль розрахунків: вибрати джерело світла та світильник, визначити кількість світильників і розташувати їх на плані приміщення з відомими геометричними розмірами. Розрахунки загального рівномірного штучного освітлення для суспільних приміщень звичайно виконується методом світлового потоку (коефіцієнту використання), при цьому допускається відхилення розрахункової освітленості від нормованої не більше ніж на $-10 \div +20\%$.

Метод світлового потоку, як правило, використовують для розрахунку потужності освітлювальної установки при рівномірному розміщенні світильників загального освітлення над горизонтальною площиною, коли відсутні великогабаритні затіняючі предмети.

Розрахунок освітлення методом коефіцієнта використання виконується по формулі

$$\Phi = E \cdot S \cdot k \cdot z / N \cdot \eta, \quad (4.6)$$

де Φ – необхідний світловий потік ламп у кожному світильнику, лм;

E – нормована мінімальна освітленість, лк, обумовлена по табл. 7.1 [5] відповідно до розряду зорової роботи;

k – коефіцієнт запасу, для суспільних приміщень, дорівнює 1,4;

S – освітлювана площа, m^2 ;

z – коефіцієнт мінімальної освітленості, величина якого знаходиться в межах 1,1-1,5 (при оптимальних відносинах відстані між світильниками до розрахункової висоти для енергозберігаючих ламп $z = 1,5$, для трубчатих люмінесцентних ламп $z = 1,1$);

N – число світильників у приміщенні;

η – коефіцієнт використання світлового потоку.

Коефіцієнт η визначається за світлотехнічними таблицями залежно від показника приміщення i , коефіцієнтів відбиття стін та стелі.

$$i = \frac{L \cdot S}{h(L + S)}, \quad (4.7)$$

де L і S – довжина і ширина приміщення, м;

h – висота світильника над робочою поверхнею, м.

Вихідні дані:

- довжина приміщення 12 м;
- ширина приміщення 6 м;
- нормована освітленість $E_n=200$ лк;
- коефіцієнт відбиття: стелі $\rho_{\text{п}}=70\%$,
- стін $\rho_{\text{с}}=50\%$;
- тип світильника з лампами розжарювання – “універсаль”;
- висота світильника над робочою поверхнею 2,6 м.

Визначаємо орієнтовну кількість світильників.

Рівномірність освітлення досягається при співвідношенні відстані між світильниками l і висоти їх підвісу h :

$$L=h=2,6.$$

Необхідна кількість світильників становить:

$$N = \frac{L \cdot S}{l^2} \equiv \frac{12 \cdot 6}{2,6^2} = 10 \text{ шт.}$$

Визначаємо показник приміщення:

$$i = \frac{12 \cdot 6}{2,6(12 + 6)} = 1,54.$$

Визначаємо мінімально необхідну величину світлового потоку лампи розжарювання при її використанні в світильнику типу “Універсаль”:

$$\Phi = \frac{200 \cdot 12 \cdot 6 \cdot 1,3 \cdot 1,15 \cdot 100}{1 \cdot 10 \cdot 51} = 3914 \text{ лм.}$$

За таблицею 8.2 [26] знаходимо коефіцієнт використання світлового потоку для світильника типу “Універсаль” при $I=1,54$, $\rho_p=70\%$, $\rho_c=50\%$.

Вибираємо лампу розжарювання потужністю з $P^{\text{розж}} = 300 \text{ Вт}$, світловим потоком $4600 \text{ лм} > 3914 \text{ лм}$.

Загальна потужність електроенергії, яка повинна підводитися до системи освітлення становитиме:

$$P_{\text{заг}}^{\text{розж}} = P_L^{\text{розж}} \cdot N = 300 \text{ Вт} \cdot 10 = 3000 \text{ Вт} = 3 \text{ кВт.}$$

Визначаємо світловий потік для люмінесцентної лампи, якщо освітлювальна установка буде виконана з світильників з люмінесцентними лампами (по 2 лампи на 1 світильник):

$$\Phi = \frac{200 \cdot 12 \cdot 6 \cdot 1,5 \cdot 1,15 \cdot 100}{2 \cdot 10 \cdot 51} = 2435 \text{ лм.}$$

Вибираємо люмінесцентну лампу потужністю $P^{\text{л.л.}} = 40 \text{ Вт}$, світловим потоком $2500 \text{ лм} > 2435 \text{ лм}$.

Загальна потужність в разі використання люмінесцентних ламп становитиме:

$$P_{\text{заг}}^{\text{л.л.}} = P_L^{\text{л.л.}} \cdot N \cdot n = 40 \text{ Вт} \cdot 10 \cdot 2 = 800 \text{ Вт} = 0,8 \text{ кВт.}$$

Економія електроенергії за рік при використанні люмінесцентних ламп становитиме:

$$260 \text{ дн} \cdot 10 \text{ год} \cdot (3 \text{ кВт} - 0,8 \text{ кВт}) = 5720 \text{ кВт год/рік},$$

де 260 – число робочих днів за рік;

10 год – зміна в темний час доби.

4.4 Висновки до четвертого розділу

В розділі був проведений експеримент по управлінню рухом мобільного колісного робота. В результаті експерименту був розроблений алгоритм автономного управління рухом мобільного робота. Результати, отримані завдяки тестуванню розробленого алгоритму, підтвердили прийнятність алгоритму, здатного автоматизувати пересування мобільного робота. Для реалізації алгоритму управління рухом потрібна корекція, дані з інфрачервоного датчика збиратимуться не постійно, а на певних кроках. Щоб здійснити планування маршруту переміщення мобільного робота під час його переходу від поточного положення до наступної точки маршруту, потрібно змістити кут його розвороту на місці.

На основі проведеного аналізу для використання в навчальному чотириколісному роботі обрано інфрачервоний 4-канальний датчик перешкод F233-01. Він забезпечує оптимальне співвідношення ціни, точності та функціональних можливостей для виконання завдань у невеликому закритому приміщенні. Застосування цього датчика дозволить реалізувати базові функції навігації та уникнення перешкод, що є ключовими для навчальних проєктів. Вихідний сигнал, що отримується з даного датчика, обернено пропорційний – зі збільшенням відстані його значення повільно зменшується. Для отримання формули для розрахунку відстані в залежності від напруги проведено дослід, в

якому знято залежність напруги від відстані. Наведений приклад графіку залежності напруги від відстані.

Був проведений розрахунок штучного освітлення робочого місця монтажу та експлуатації реконфігурованого мобільного робота. Вибрано люмінесцентну лампу потужністю $P^{л.л.} = 40$ Вт для освітлення робочого місця користувача.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено лабораторний макет реконфігурованого мобільного робота для навчання студентів.

У першому розділі проведений аналіз конструкцій реконфігурованих мобільних роботів. Був зроблений висновок щодо актуальності застосування реконфігурованих мобільних роботів. Проведений аналіз методів керування реконфігурованими мобільними роботами. Обрана система автоматичного регулювання. Аналіз вибору протоколу передачі даних показав, що протокол Bluetooth Low Energy є найбільш відповідним для реалізації поставленого завдання.

У другому розділі представлена загальна структурна схема для реконфігурованого мобільного робота та системи керування ним, а також функціональна схема, що відображає функції окремих елементів системи і зв'язки між ними. Був проведений аналіз апаратних модулів та вибрані мікроконтролер Arduino, серводвигуни SG90, Bluetooth передавачі HC-05. На основі розроблених схем підключення апаратних модулів реконфігурованого мобільного робота було проведено складання макету робота.

У третьому розділі була проведена розробка програмного забезпечення керування мобільним роботом. Було вибрано середовище розробки, розроблені алгоритми роботи всіх систем реконфігурованого мобільного робота. У процесі написання коду проекту замість стандартних бібліотек було вирішено написати свій клас Motor для роботи з двигунами, що спростило функціонал, а власний код дав більше контролю та легкість розуміння. Розроблений приклад програмної реалізації функцій String та часу. Ці функції дозволяють організувати програму практично будь-якої складності з будь-якою кількістю завдань, що паралельно виконуються за таймером.

У четвертому розділі проведений експеримент з розробки алгоритму

автономного управління рухом мобільного робота. Розглянуто переваги та недоліки датчиків відстані, на основі аналізу вибраний інфрачервоний датчик.

Актуальність роботи та її практичне застосування полягає в тому, що перепрофілюючи колісного робота в реконфігурованого мобільного робота з набором різних модулів і можливістю перепрограмування для виконання різних завдань, можливо створити доступну робототехніку, Такий робот може бути використаний у навчальних закладах для вивчення основ робототехніки та програмування.

Мета роботи – підвищення ефективності проведення освітнього процесу досягнута за рахунок написання свого класу Motor для роботи з двигунами, що спростило функціонал, а власний код дав більше контролю та легкість розуміння роботи реконфігурованого мобільного робота, та у використанні корпуса для електроніки з вентиляцією, до якого за допомогою конекторів jack 3,5 мм підключаються всі модулі, що зменшує час на реконфігурацію моделі мобільного робота.

Наукова цінність проєкту полягає в розробці, дослідженні та впровадженні ефективного алгоритму управління рухом мобільного робота, який поєднує в собі базові принципи робототехніки та елементи інтелектуальної навігації.

В результаті використання реконфігурованого мобільного робота під час занять з робототехніки у студентів формуються уміння:

- планувати навчальний процес з освітньої робототехніки та робототехнічних систем;
- розроблювати та проводити уроки з освітньої робототехніки та робототехнічних систем; готувати відповідні навчально-методичні матеріали; проводити аналіз проведених уроків;

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 4 "Якісна освіта", а саме п. 4.4 "Підвищити якість вищої освіти та забезпечити її тісний зв'язок з наукою, сприяти формуванню в країні міст освіти та науки".

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення [Текст]. – Введ. 2015-06-22. – К.: Держстандарт України, 2017., 29 с.

2. Методичні вказівки з підготовки кваліфікаційної роботи магістра для студентів усіх форм навчання спеціальності 174 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, А.О.

3. Бучинський М. Я., Горик О. В., Чернявський А. М., Яхін С. В. Основи творення машин / [За редакцією О. В. Горика, доктора технічних наук, професора, заслуженого працівника народної освіти України]. – Харків: Вид-во «НТМТ», 2017. – 448 с.

4. Виробничі процеси та обладнання об'єктів автоматизації [Текст]: Підручник для студентів вищих навчальних закладів / І. Ш. Невлюдова. Кривий Ріг: Криворізький коледж НАУ, 2017 р. – 444 с. [Електронний ресурс] : [Вебсайт]. URL: <https://openarchive.nure.ua/bitstream/document/9556/3/Pidruchnyk.PDF>

5. Денисюк В.Ю. Електронний посібник з дисципліни «Автоматизація виробничих процесів в приладобудуванні»./ Денисюк В.Ю. – Луцьк: Луцький національний технічний університет, 2017. URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%90%D0%92%D0%9F%20%D0%B2%20%D0%9F%D0%91%20PDF%201/other/tema_8__zastosuvannya_promislovix_robotiv_i_gnuchkix_virobnichix_sistem_dlya_avtomatizacziyi_virobnictva_v_priladobuduvanni.pdf.

6. Інтерфейс послідовного модуля Bluetooth HC-05 на основі AT-команд із Minicom у Linux. [Електронний ресурс] URL:

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=494cff910ec55ae4fc1f17c535297ece4580fb54>.

7. Невлюдов І. Ш. Комп'ютерно-інтегровані технології виробництва технічних засобів автоматизації. Частина 1: підручник для студентів закладів вищої освіти ; Харків. нац. ун-т радіоелектроніки. – Харків : ФОП Панов А.М., 2021. – 604 с. ISBN 978-617-7947-67-6.

8. Невлюдов І.Ш. Виробничі процеси та обладнання об'єктів автоматизації: Підручник / Кривий Ріг: КК НАУ, 2017. – 444 с.

9. Невлюдов І.Ш. Основи наукових досліджень: Навч. посібник /І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривий Ріг: Криворізький коледж НАУ, 2019. – 396 с.

10. Невлюдов І.Ш. Людино-машинний інтерфейс в технічних засобах автоматизації: Навчальний посібник / І.Ш. Невлюдов, О.І. Филипенко, Б.О. Шостак. – Харків : «ХТМТ», 2019. – 244 с.

11. Невлюдов І.Ш., Токарева О.В. Теорія автоматичного управління (збірник задач) [Текст]: навч. посіб. для студентів спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології / І.Ш. Невлюдов, О.В. Токарева; Харків. нац. ун-т радіоелектроніки. - Харків: Панов А.М., 2020. – 240 с.

12. Цвіркун Л.І. Робототехніка та мехатроніка: навч. посіб. / Л.І. Цвіркун, Г. Грулер ; під заг. ред. Л.І. Цвіркуна ; М-во освіти і науки України, Нац. гірн. ун-т. – 3-тє вид., переробл. і доповн. – Дніпро: НГУ, 2017. – 224 с.

13. Springer Handbook of Automation / Ed. by S. Y. Nof. – Berlin: Springer Verlag, 2009. – 1812 с.

14. Вибір бездротової технології передавання даних для обладнання навчальних лабораторій [Електронний ресурс] URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2017/oct/6570/003-010-016.pdf>

15. Сервоприводы Ардуино SG90, MG995: схема подключения и управление [Електронний ресурс] URL: <https://arduinomaster.ua/motor-dvigatel-privod/servoprivody-arduino-sg90-mg995-shema-podklyuchenie-upravlenie>
16. Шилдт Г. С++. Базовий курс / Герберт Шилдт. – Київ: Вільямс, 2015. – 602 с. – (3).
17. Гантерот К. Оптимізація програм на С++ / Курт Гантерот. – Київ: Вільямс, 2013. – 400 с.
18. Петін В. 77 преєктів для Arduino / Віктор Петін. – Київ: ДМК-Пресс, 2012. – 352 с.
19. .Ranberg D. Arduino for inventors / Derek Ranberg. – London: Usborne, 2014. – 288 с.
20. .Knight S. Arduino for Beginners: Step-by-Step Guide to Arduino / Simon Knight. – Washington: Independently published, 2012. – 139 с
21. Metal Spheres Swarm Together to Create Freeform Modular Robots [Електронний ресурс] Режим доступу: <https://spectrum.ieee.org/freebots-spheres-swarm-robots>
22. SnailBot: A Continuously Dockable Modular Self-reconfigurable Robot Using Rocker-bogie Suspension [Електронний ресурс] URL: https://freeformrobotics.org/wp-content/uploads/2022/03/ICRA22_0763_FI.pdf
23. Mori: A Modular Origami Robot [Електронний ресурс] URL: <https://ieeexplore.ieee.org/document/7907345>
24. IR датчик відстані Sharp GP2Y0A21YK0F аналоговий [Електронний ресурс] URL: <https://arduino.ua/prod4626-ir-datchik-rasstoyaniya-sharp-gp2y0a21yk0f-analogovii-10-80-sm-ot-pololu>
25. ДБН В.2.5-28-2006 Державні будівельні норми. Природне і штучне освітлення.