

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

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

**МІНІ-РОБОТ НА БАЗІ ПЛАТИ BBC MICRO:BIT
MINI ROBOT BASED ON BBC MICRO:BIT BOARD**

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42
Фаєр Марк Тарасович

(підпис)

Керівник:
к.т.н., доцент
Поліщук Микола Миколайович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 04 » червня 2025 р.

Гарант освітньої програми:

к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т. Терлецький

« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Фасру Марку Тарасовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Міні-робот на базі плати BBC micro:bit

Керівник роботи к.т.н., доцент Поліщук Микола Миколайович

затверджені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

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

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

Вступ

Аналіз технологій для розробки міні-роботів

Апаратна частина міні-робота

Розробка апаратного та програмного забезпечення міні-робота

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз технологій для розробки міні-роботів</i>	<i>Поліщук М.М., доцент</i>		
<i>Апаратна частина міні-робота</i>	<i>Поліщук М.М., доцент</i>		
<i>Розробка апаратного та програмного забезпечення міні-робота</i>	<i>Поліщук М.М., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	_____ %		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст. викладач</i>		

7. Дата видачі завдання 10.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	до 10.02.2025 р.	Виконано
2.	<i>Вибір апаратної та програмної бази для проєкту</i>	до 02.03.2025 р.	Виконано
3.	<i>Проєктування та тестування міні-робота</i>	до 02.04.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків</i>	до 02.05.2025 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
8.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 10.05.2025 р.	Виконано
9.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
10.	<i>Інструментальна перевірка на академічний плагіат</i>	до 04.06.2025 р.	Виконано
11.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедру</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

(підпис)

Фаєр М.Т.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Поліщук М.М.

(прізвище, ініціали)

АНОТАЦІЯ

Фаєр М. Т. Міні-робот на базі плати BBC micro:bit. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

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

Перший розділ присвячено огляду предметної області, тут обґрунтовано актуальність обраної теми, здійснено огляд платформ для створення міні-роботів, представлено характеристики можливостей плати BBC micro:bit, проаналізовано сенсори та приводи для міні-роботів та протоколи взаємодії між компонентами.

В другому розділі здійснено вибір компонентів для апаратної реалізації, а саме плата BBC micro:bit як основа управління, сенсори для орієнтації, приводи для забезпечення руху робота. Представлено схемотехнічне рішення для інтеграції компонентів та особливості програмування BBC micro:bit.

Третій розділ присвячено огляду комплектуючі, проектування та складання міні-робота, а також представлено програмування плати BBC micro:bit для дистанційного керування міні-роботом.

Ключові слова: дистанційне керування, сервопривід, micro:bit, міні-робот, складання, програмування.

ANNOTATION

Fayer M. Mini-robot based on the BBC micro:bit board. Manuscript.

Qualification work of the bachelor of the specialty «Computer Engineering» of the specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

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

The first section is devoted to an overview of the subject area, the relevance of the chosen topic is substantiated, platforms for creating mini-robots are reviewed, the characteristics of the capabilities of the BBC micro:bit board are presented, sensors and drives for mini-robots and interaction protocols between components are analyzed.

In the second section, the selection and justification of the choice of components for hardware implementation are made, namely the BBC micro:bit board as the basis for control, sensors for orientation, drives for ensuring the movement of the robot. A circuit solution for integrating components and features of BBC micro:bit programming are presented.

The third section is devoted to an overview of the components, design and assembly of the mini-robot, and also presents the programming of the BBC micro:bit board for remote control of the mini-robot.

Keywords: remote control, servo, micro:bit, mini-robot, assembly, programming.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ МІНІ-РОБОТІВ.....	8
1.1 Огляд платформ для створення міні-роботів	8
1.2 Характеристика можливостей плати BBC micro:bit.....	12
1.3 Аналіз сенсорів і приводів для міні-роботів	15
1.4 Протоколи взаємодії між компонентами (Bluetooth, I2C, SPI)	18
РОЗДІЛ 2 АПАРАТНА ЧАСТИНИ МІНІ-РОБОТА	21
2.1 Обґрунтування вибору компонентів для апаратної реалізації	21
2.1.1 Плата BBC micro:bit як основа управління.....	21
2.1.2 Сенсори для орієнтації та виявлення перешкод	22
2.1.3 Приводи для забезпечення руху робота	23
2.2 Схемотехнічне рішення для інтеграції компонентів.....	25
2.3 Особливості програмування BBC micro:bit: середовища та інструменти .	28
РОЗДІЛ 3 РОЗРОБКА АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МІНІ-РОБОТА.....	31
3.1 Комплектуючі міні-робота.....	31
3.2 Складання міні-робота.....	34
3.3 Програмування плати BBC micro:bit для дистанційного керування міні-роботом	40
ВИСНОВКИ.....	45
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТКИ.....	50

ВСТУП

У сучасному світі робототехніка стрімко інтегрується в різноманітні сфери життя й особливу роль відіграє розвиток міні-роботів як функціональних платформ, що дозволяють ефективно вивчати основи програмування, електроніки та автоматизації. Однією з найпопулярніших платформ є мікроконтролерна плата BBC micro:bit.

Проектування міні-робота на базі BBC micro:bit є не лише технічно доцільним, а й актуальним з освітньої, дослідницької та практичної точки зору. Така система може стати основою для подальших розробок у сфері автономних пристроїв, IoT-рішень та мобільної робототехніки.

Метою роботи є реалізація міні-робота на базі плати BBC micro:bit з можливістю дистанційного керування.

Об'єкт дослідження – мініатюрна робототехнічна система, побудована на базі мікроконтролерної плати BBC micro:bit.

Предмет дослідження – апаратна реалізація та програмування міні-робота.

У ході дослідження для побудови міні-робота на базі BBC micro:bit було поставлено такі основні завдання:

- здійснити порівняльний аналіз сучасних апаратних платформ для розробки міні-роботів;
- охарактеризувати технічні можливості мікроконтролера BBC micro:bit;
- обґрунтувати вибір апаратних компонентів для створення міні-робота;
- зібрати та протестувати міні-робота.

Практична значимість розробленого міні-робота полягає в можливості його використання для опанування основ мікроконтролерного програмування, схемотехніки та робототехніки. Завдяки відкритій архітектурі, міні-робот може бути адаптований під різні дослідницькі цілі.

РОЗДІЛ 1

АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ МІНІ-РОБОТІВ

1.1 Огляд платформ для створення міні-роботів

Розробка міні-роботів є однією з найдинамічніших галузей комп'ютерної інженерії, що об'єднує мікроелектроніку, механіку, програмування та бездротові технології. Основу створення таких пристроїв становлять спеціалізовані платформи, які забезпечують необхідний набір апаратних ресурсів і програмних можливостей для реалізації проєктів різної складності.

Однією з найпоширеніших платформ є Arduino [1], зокрема плати типу Arduino Uno, Nano та Mega. Arduino вирізняється простотою використання, великою кількістю готових бібліотек та модулів, що робить її ідеальною для початківців та освітніх проєктів. Крім того, Arduino має відкриту архітектуру, що дозволяє розробникам змінювати та адаптувати апаратне забезпечення відповідно до потреб. Водночас для побудови міні-роботів часто використовуються додаткові щити (shields) для керування моторами, зчитування сенсорних даних або забезпечення бездротової комунікації. Наприклад, у проєкті створення автономного робота з сенсорами зіткнення (рис. 1.1), використовується Arduino для керування рухом та обробки даних з датчиків [2].

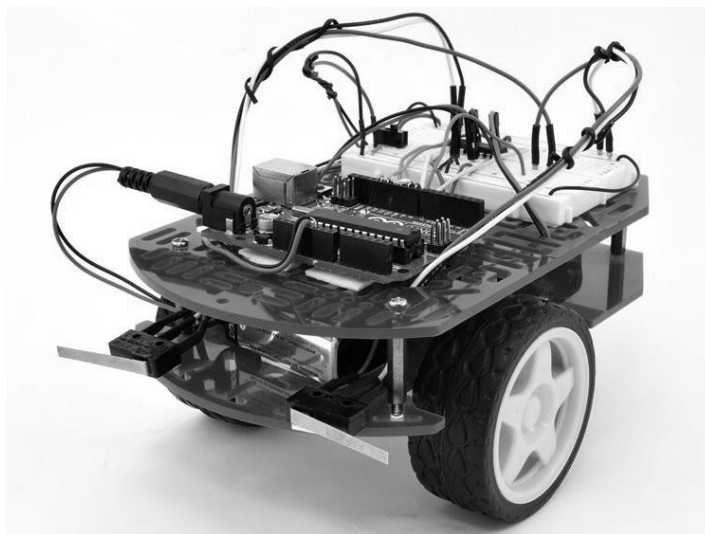


Рисунок 1.1 – Робот з двома датчиками удару, встановленими на передній панелі [2]

Іншою популярною платформою є Raspberry Pi [3], яка, на відміну від Arduino, є повноцінним одноплатним комп'ютером. Raspberry Pi дозволяє реалізовувати складні алгоритми обробки зображень, машинного навчання та навігації, що є важливим для автономних міні-роботів. Однак Raspberry Pi має вищі енергетичні потреби і складніше налаштування системи реального часу, що може бути недоліком для деяких мініатюрних проєктів. Наприклад, у дослідженні 2020 року було представлено методологію планування маршруту та уникнення перешкод для автономних роботів з використанням Raspberry Pi [4].

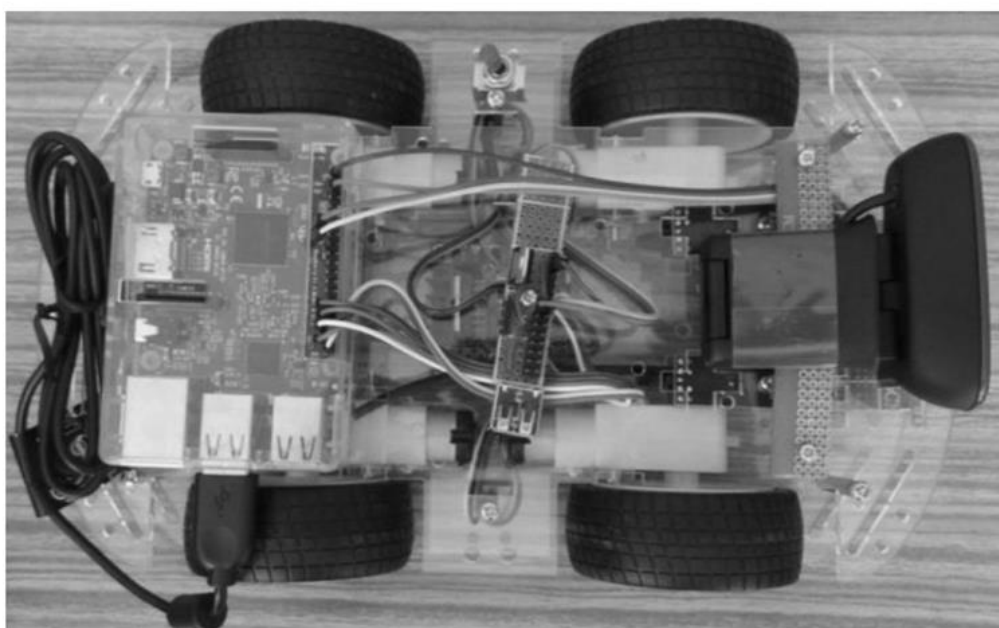


Рисунок 1.2 – Апаратне оформлення навігаційного модуля [4]

Особливої уваги заслуговує BBC micro:bit [5], платформа, спеціально створена для освітніх цілей. Плата має вбудовані акселерометр, магнітометр, Bluetooth-інтерфейс, матрицю світлодіодів 5x5 і кнопки керування. Завдяки простому програмуванню на JavaScript, Python або за допомогою блокових середовищ (наприклад, Microsoft MakeCode) micro:bit є зручним інструментом для навчання основам робототехніки. Саме завдяки вбудованим модулям і можливості бездротового підключення BBC micro:bit широко використовується

у проєктах створення міні-роботів, зокрема мобільних платформ, роботів-маніпуляторів та інтерактивних ігрових систем.

У 2025 році було представлено новий інструмент MicroCode, який дозволяє програмувати micro:bit безпосередньо на пристрої за допомогою візуального середовища програмування (рис. 1.3).

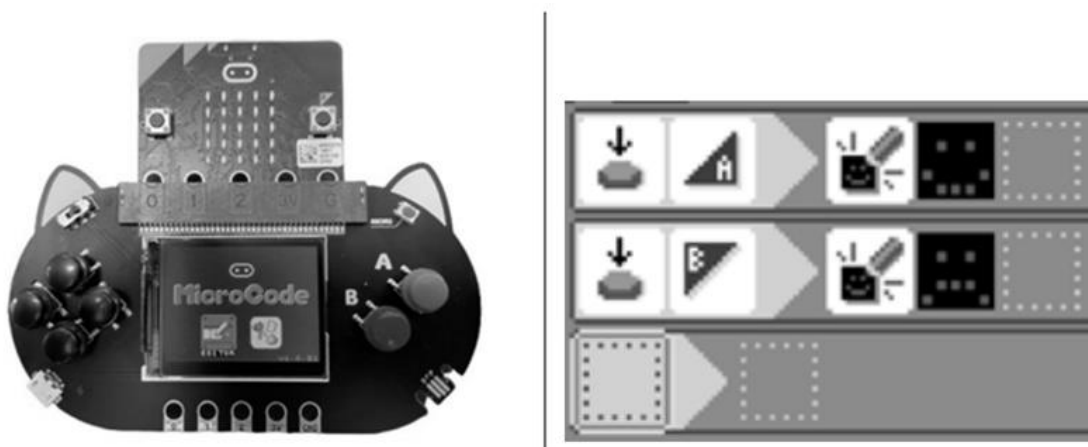


Рисунок 1.3 – Micro:bit V2 вставлений у щит MicroCode і запущений Мікрокод; скріншот редактора MicroCode, на якому відображається програма [6]

Ще однією перспективною платформою є ESP32 [7], яка поєднує функції мікроконтролера та бездротового комунікатора з підтримкою Wi-Fi та Bluetooth. ESP32 широко застосовується для створення інтернет-керованих міні-роботів, систем моніторингу та автономних агентів у мережах IoT. Завдяки двоядерній архітектурі та низькому енергоспоживанню ESP32 дозволяє реалізовувати складні обчислювальні задачі, залишаючись ефективним рішенням для невеликих робототехнічних проєктів. Наприклад, у дослідженні [8] було представлено систему роботизованого письма з інтеграцією штучного інтелекту, яка використовує Raspberry Pi Pico (на базі ESP32) для відтворення рукописного тексту з високою точністю.

Серед освітніх рішень варто також відзначити платформу Tinkercad Circuits [9], яка дозволяє моделювати роботу електронних схем та мікроконтролерів у віртуальному середовищі. Хоча ця платформа

використовується переважно для симуляцій, вона суттєво полегшує процес навчання базовим принципам електроніки та робототехніки.

Окрему групу становлять готові робототехнічні набори на базі зазначених платформ, зокрема:

- mBot на основі Arduino для навчання основ програмування і робототехніки [10];
- PiCar-S на базі Raspberry Pi для вивчення автономного водіння [11];
- Maqueen на базі BBC micro:bit для початкового знайомства з робототехнікою [12].

Порівнюючи платформи між собою, можна виділити такі основні критерії вибору для створення міні-роботів:

- простота програмування, де BBC micro:bit та Arduino мають інтуїтивно зрозуміле програмне середовище, що підходить для початківців;
- функціональні можливості, оскільки Raspberry Pi та ESP32 забезпечують більшу обчислювальну потужність і мережеву інтеграцію;
- вартість, адже BBC micro:bit і Arduino є найбільш доступними для широкого кола користувачів;
- енергоспоживання, оскільки BBC micro:bit та ESP32 відзначаються низьким споживанням енергії, що є особливо важливим для мобільних міні-роботів.

Вибір платформи для створення міні-робота значною мірою визначається конкретними цілями проєкту. Наприклад, для навчальних завдань оптимальним вибором є BBC micro:bit, оскільки він поєднує доступність, простоту інтеграції з додатковими модулями та широкі можливості для реалізації проєктів початкового та середнього рівня складності.

Таким чином, огляд сучасних платформ показує, що розвиток міні-робототехніки базується на постійній еволюції апаратних рішень, що спрощують процес створення та налаштування роботизованих систем. Саме тому правильний вибір платформи є одним із ключових етапів у процесі розробки ефективного і функціонального міні-робота.

1.2 Характеристика можливостей плати BBC micro:bit

Плата BBC micro:bit є однією з найбільш доступних та функціональних платформ для навчання основ програмування, електроніки та створення міні-роботів. Спочатку розроблена британською компанією BBC у співпраці з ARM, Microsoft та іншими технологічними гігантами, ця плата орієнтована на освітнє використання та проєктну діяльність у сфері STEM-освіти [5].

BBC micro:bit є компактним мікроконтролерним пристроєм розміром 4×5 см, який містить низку інтегрованих модулів (рис. 1.4).

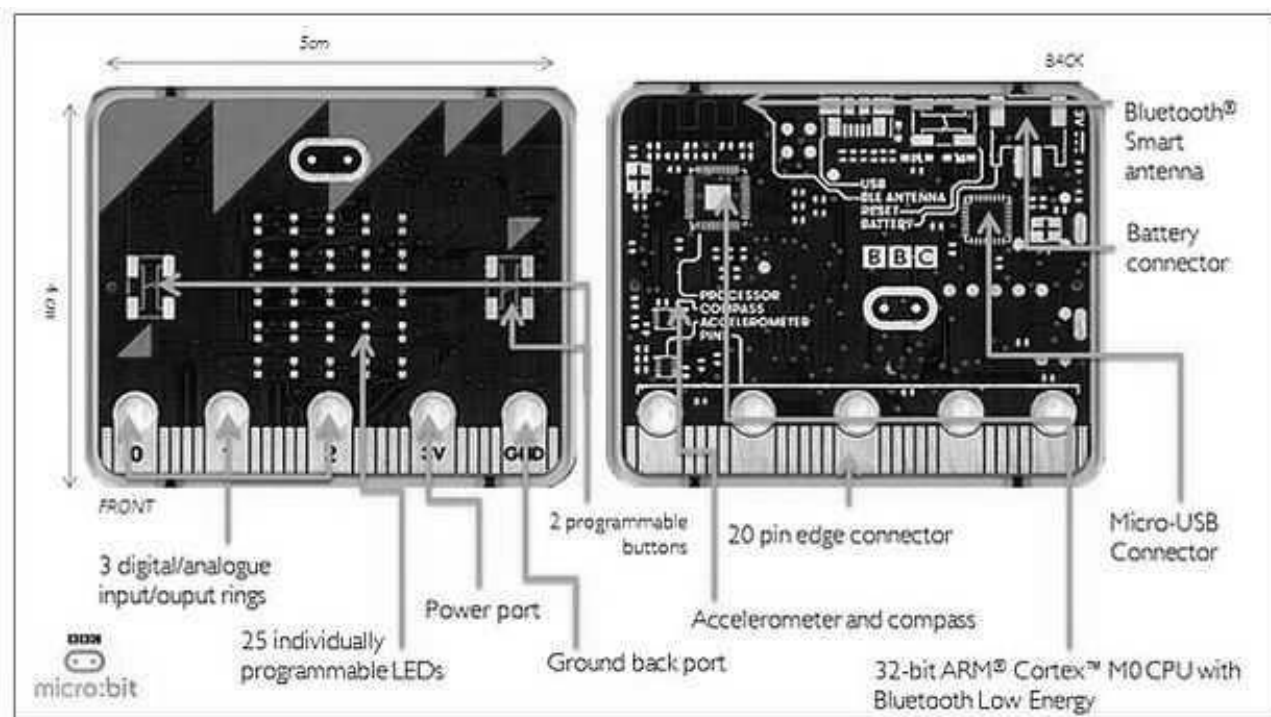


Рисунок 1.4 – Структурна схема плати BBC micro:bit [5]

ARM Cortex-M0 є енергоефективним 32-бітовим процесором, який забезпечує базову обчислювальну потужність для обробки вхідних даних, виконання алгоритмів і керування периферійними пристроями. Завдяки своїй архітектурі він дозволяє працювати з низьким енергоспоживанням, що є критичним для мобільних і автономних пристроїв.

Інтегрована матриця зі світлодіодів 5×5 використовується для виведення базової графічної або текстової інформації. Вона дозволяє відображати повідомлення, символи, прості графічні зображення та служить засобом візуальної індикації стану пристрою.

Акселерометр, вбудований у плату, дозволяє визначати орієнтацію micro:bit у просторі, реєструвати рухи, удари та нахили. Це розширює можливості пристрою для розробки проєктів, пов'язаних з управлінням жестами або детекцією рухів.

Вбудований магнітометр функціонує як електронний компас, що дозволяє визначати напрямок на основі магнітного поля Землі. Це дає змогу реалізовувати навігаційні проєкти або застосування, пов'язані з орієнтацією у просторі.

Bluetooth Low Energy (BLE) модуль забезпечує бездротовий обмін даними з іншими пристроями. Завдяки цьому micro:bit може взаємодіяти зі смартфонами, планшетами або іншими платами, що дозволяє створювати мережі пристроїв та IoT-системи.

На платі передбачено дві кнопки (А та В), які можна програмно обробляти. Вони дозволяють користувачу взаємодіяти з пристроєм у режимі реального часу, що є важливим для розробки інтерактивних застосувань.

Контакти введення/виведення дають можливість підключення зовнішніх сенсорів, моторів, світлодіодів та інших електронних компонентів. Це дозволяє розширювати базові функціональні можливості плати та будувати складні робототехнічні системи.

Завдяки інтеграції цих компонентів micro:bit дозволяє реалізовувати широкий спектр проєктів без необхідності в додаткових модулях або складних налаштуваннях.

Мережеві можливості micro:bit, зокрема підтримка Bluetooth Low Energy (BLE), відкривають шлях до створення інтерактивних бездротових проєктів. Користувачі можуть об'єднувати кілька плат у мережу для обміну повідомленнями, відправлення команд або створення IoT-проєктів невеликого масштабу [13].

Щодо апаратних розширень, BBC micro:bit підтримує підключення зовнішніх компонентів через роз'єм із 20 контактами. Це дозволяє під'єднувати мотор-контролери, сервоприводи, датчики температури, вологості, освітленості, RFID-модулі та інші периферійні пристрої.

Крім того, існують спеціальні платформи, такі як Kitronik Motor Driver Board та DFRobot Maqueen, що оптимізують підключення та керування приводами для створення мобільних роботів.

У плані енергоефективності micro:bit споживає дуже мало енергії, для роботи може використовувати батарейки типу AAA або живлення через USB-порт. У новій версії BBC micro:bit V2 (рис. 1.5) було додано покращення енергоспоживання завдяки модернізованому контролеру енергоживлення.

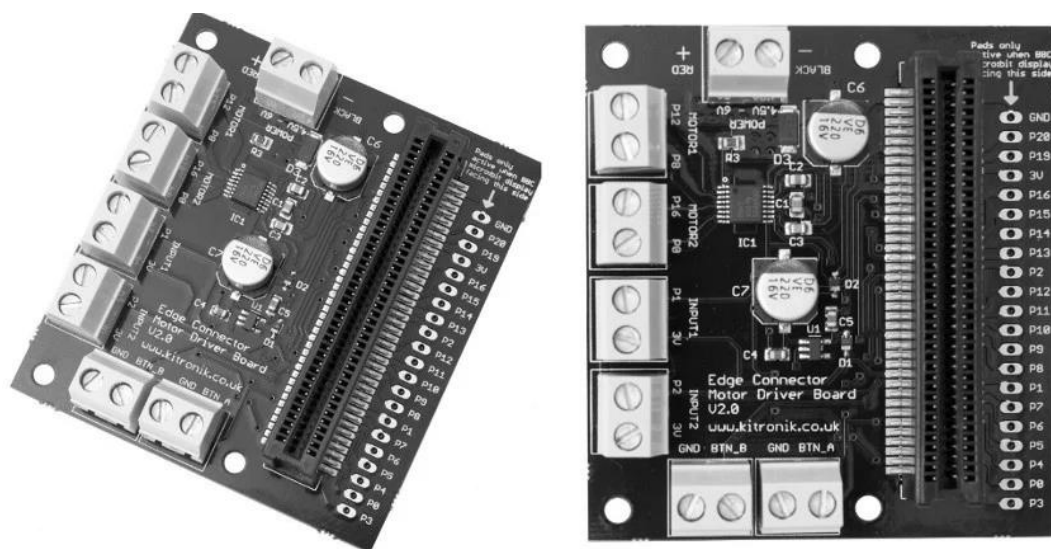


Рисунок 1.5 – Плата драйвера двигуна для BBC micro:bit V2 [5]

Серед важливих особливостей можна також виокремити:

- можливість використання модуля радіозв'язку для бездротового передавання даних між платами без необхідності налаштування мережі Wi-Fi;
- можливість використання режиму low-power sleep для зменшення енергоспоживання у стані очікування;
- підтримку великої кількості готових бібліотек для швидкої розробки проєктів.

Таким чином, BBC micro:bit є ідеальним вибором для створення міні-роботів завдяки простоті використання, вбудованим сенсорам, підтримці бездротових технологій та широким можливостям розширення.

1.3 Аналіз сенсорів і приводів для міні-роботів

Функціональність міні-робота значною мірою визначається типами сенсорів і приводів, які використовуються для взаємодії з навколишнім середовищем. Платформа BBC micro:bit надає достатньо можливостей для підключення та керування різноманітними сенсорами та приводними пристроями через свої I/O-піни, а також завдяки вбудованим сенсорам.

У процесі проектування міні-роботів на базі плати BBC micro:bit важливу роль відіграє правильний підбір сенсорних модулів, які забезпечують взаємодію з навколишнім середовищем. Вибір конкретного сенсора залежить від поставленого завдання, необхідної точності, умов експлуатації та сумісності з апаратною платформою. У таблиці 1.1 узагальнено найбільш поширені типи сенсорів, що можуть використовуватися з micro:bit.

Таблиця 1.1 – Сенсори для міні-роботів на базі BBC micro:bit [14-16]

Назва сенсора	Тип / категорія	Принцип дії	Призначення / опис	Сумісність з micro:bit
1	2	3	4	5
вбудований акселерометр	інерціальний датчик	вимірює прискорення по осях X, Y, Z	визначає рух, положення, удари, нахили	вбудований
вбудований магнітометр	магнітний датчик	визначає напрямок у магнітному полі Землі	орієнтація у просторі, компас	вбудований
HC-SR04	ультразвуковий	вимірює час повернення ультразвукового імпульсу від об'єкта	вимірювання відстані до перешкод	через адаптер
TCRT5000	інфрачервоний	визначає відбитий інфрачервоний промінь	слідкування за лінією, детекція перешкод	так

Продовження таблиці 1.1

1	2	3	4	5
TCS34725	оптичний / кольоровий	аналізує спектр відбитого світла	розпізнавання кольору, сортування об'єктів	так (через I2C)
DHT11 / DHT22	температурний / гігрометр	зчитує вологість і температуру за зміною електричного опору	контроль клімату, метеостанції	так
BH1750	датчик освітленості	генерує сигнал пропорційно до рівня освітлення	автоматичне регулювання яскравості, адаптація до освітлення	так (I2C)
MQ-2 / MQ-135	газовий датчик	зміна провідності чутливого шару при наявності газів	виявлення CO, CH ₄ , диму, небезпечних речовин	так (аналоговий вхід)

Аналіз наведених у таблиці сенсорів свідчить про широкі можливості розширення функціональності міні-роботів на базі BBC micro:bit. Завдяки вбудованим сенсорам (акселерометру, магнітометру) та підтримці зовнішніх модулів (ультразвукових, інфрачервоних, температурних, оптичних тощо), платформа забезпечує повний спектр сенсорного сприйняття, необхідного для створення навчальних, дослідницьких і прикладних робототехнічних систем.

Побудова розподілених сенсорних мереж із використанням плати BBC micro:bit відкриває широкі можливості для створення модульних робототехнічних систем із підвищеною гнучкістю й масштабованістю. У своєму дослідженні Р. Дубань та Ю. Юрко наголошують на тому, що використання розподіленої архітектури дає змогу кожному сенсорному модулю працювати автономно, збираючи дані у своїй зоні відповідальності, після чого інформація агрегується центральним процесором або передається між вузлами для децентралізованої обробки. Зокрема, реалізація сенсорних мереж на основі micro:bit дозволяє спрощувати апаратну структуру роботів і збільшувати їх автономність завдяки мінімізації комунікаційних затримок і енергетичних втрат [17].

Варто зауважити, що приводи також є невід’ємною складовою будь-якого міні-робота, оскільки саме вони забезпечують виконання фізичних дій – переміщення, обертання, позиціонування тощо. Плата BBC micro:bit підтримує підключення різних типів виконавчих пристроїв завдяки доступним інтерфейсам керування, включно з ШІМ-сигналами та цифровими пін-контактами.

У статті [18] висвітлюється використання micro:bit для управління різними приводами у складі автономної робототехнічної системи. Це доводить здатність плати micro:bit ефективно взаємодіяти з приводними пристроями через прості інтерфейси керування, що має безпосереднє відношення до створення автономних мобільних систем на її основі, зокрема міні-роботів, які потребують керування рухом або маніпулюванням об’єктами. У таблиці 1.2 систематизовано найбільш поширені типи приводів, що використовуються в міні-роботах.

Таблиця 1.2 – Типи приводів для міні-роботів на базі BBC micro:bit [18]

Тип приводу	Принцип дії	Призначення / опис	Переваги	Сумісність з micro:bit
серводвигун (SG90)	керується ШІМ сигналом для встановлення точного кута	поворотні механізми, маніпулятори, сенсорні платформи	висока точність, легке керування	пряме підключення (через пін + GND + PWM)
двигун постійного струму (DC motor)	обертання валу залежно від напруги живлення та полярності	привід коліс у мобільних платформах	простота та дешевизна	через драйвер (L298N, TB6612FNG)
кроковий двигун	обертання з фіксованим кроком через серію імпульсів	позиціонування, точне переміщення	висока точність без зворотного зв’язку	через драйвер з бібліотекою
вібромотор	генерує вібрацію при подачі напруги	сигнальні індикатори, прості тактильні відповіді	малий розмір, низьке споживання	пряме підключення
Maqueen шасі (вбудовані мотори)	керується через вбудований драйвер на платі розширення	колісні платформи з поворотом на місці	інтеграція з сенсорами, простота монтажу	повна апаратна сумісність

Розгляд типів приводів свідчить про здатність плати BBC micro:bit забезпечувати ефективне керування як простими, так і точними виконавчими механізмами. Від стандартних двигунів постійного струму до сервоприводів і

інтегрованих колісних платформ, micro:bit демонструє гнучкість у реалізації різних моделей руху.

Таким чином, BBC micro:bit, попри свої компактні розміри, підтримує інтеграцію широкого спектра сенсорів і приводів. Базові вбудовані датчики значно спрощують початок проектування, а підтримка зовнішніх компонентів дозволяє поступово розширювати функціональність робота.

1.4 Протоколи взаємодії між компонентами (Bluetooth, I2C, SPI)

Функціональність сучасних міні-роботів залежить не лише від кількості інтегрованих сенсорів і приводів, а й від ефективної організації обміну даними між компонентами системи. Під час розробки міні-робота на базі BBC micro:bit важливо правильно обрати відповідний протокол комунікації, що забезпечить надійність, швидкість та енергоефективність передачі інформації. Серед найбільш релевантних стандартів для micro:bit виділяють Bluetooth Low Energy (BLE), I2C та SPI, кожен з яких має свої особливості застосування у робототехнічних проєктах.

BBC micro:bit має вбудовану підтримку стандарту BLE, що дозволяє обмінюватися даними з іншими пристроями без використання дротів. BLE характеризується мінімальним енергоспоживанням, завдяки чому особливо добре підходить для мобільних та автономних систем, де важливо зберігати заряд батареї.

Використання BLE у навчальних робототехнічних платформах на кшталт micro:bit сприяє реалізації таких можливостей:

- бездротового дистанційного керування міні-роботом через смартфон чи комп'ютер;
- координації групи роботів шляхом обміну командами;
- передачі телеметрії або сенсорних даних у реальному часі [19].

Особливістю BLE є можливість організації простих однорангових мереж без потреби в централізованому сервері, що оптимізує розробку розподілених робототехнічних систем.

Протокол I2C (Inter-Integrated Circuit) забезпечує синхронний двонаправлений обмін даними між мікроконтролерами та периферійними пристроями через дві спільні лінії: SDA (дані) та SCL (тактування).

У контексті міні-роботів на базі micro:bit I2C дозволяє:

- підключати одночасно кілька сенсорних модулів (акселерометрів, датчиків освітленості, температури тощо);
- мінімізувати кількість дротових з'єднань, що критично важливо для компактних мобільних систем;
- працювати у режимі як ведучого (Master), так і підлеглого (Slave) пристрою.

Дослідження [20] підкреслює, що I2C завдяки простоті інтеграції та низьким апаратним вимогам залишається одним із найефективніших рішень для побудови сенсорних мереж у мініатюрних робототехнічних системах.

У проєктах на базі micro:bit I2C часто використовується для підключення модулів розширення портів введення-виведення, екранів для виведення інформації та багатофункціональних сенсорних плат.

SPI – це інший синхронний протокол, який базується на чотирьох лініях (MISO, MOSI, SCK, SS) та дозволяє здійснювати швидкий обмін великими обсягами даних [5].

Основними перевагами SPI є вища швидкість передачі інформації у порівнянні з I2C, простота побудови протоколу обміну при роботі з одним або невеликою кількістю пристроїв, а також можливість використовувати SPI у застосуваннях, де потрібна мінімальна затримка.

Як відзначено в огляді [5], SPI особливо добре підходить для підключення до micro:bit високошвидкісних пристроїв, таких як пам'ять, дисплеї або сенсори реального часу.

У практиці створення міні-роботів SPI застосовується для:

- керування швидкими цифровими модулями;
- передачі даних між процесорами;
- роботи з камерами або високочутливими сенсорами руху.

У таблиці 1.3 наведено основні характеристики та приклади використання найбільш поширених протоколів обміну інформацією, підтримуваних платформою BBC micro:bit.

Таблиця 1.3 – Протоколи обміну даними між компонентами у міні-роботах на базі BBC micro:bit [19, 20]

Протокол	Тип передавання	Основні особливості	Переваги	Приклади використання
Bluetooth Low Energy (BLE)	бездротовий	низьке енергоспоживання, робота на короткі відстані	мобільність, простота інтеграції зі смартфонами	дистанційне керування, обмін даними між micro:bit
I2C (Inter-Integrated Circuit)	дротовий (синхронний)	дві загальні лінії для обміну даними між багатьма пристроями	мінімізація кількості з'єднань, простота масштабування	підключення сенсорів, екранів, розширень
SPI (Serial Peripheral Interface)	дротовий (синхронний)	чотири окремі лінії для швидкої передачі даних	висока швидкість обміну, мінімальні затримки	передача даних до швидких сенсорів, пам'яті, дисплеїв

Отже, підтримка micro:bit трьох основних протоколів обміну даними дає змогу гнучко налаштовувати архітектуру міні-робота відповідно до специфіки завдання. Використання BLE дозволяє реалізувати бездротову взаємодію між пристроями, I2C забезпечує простоту інтеграції великої кількості сенсорів, тоді як SPI надає можливість здійснювати високошвидкісний обмін даними для більш складних застосувань.

РОЗДІЛ 2

АПАРАТНА ЧАСТИНИ МІНІ-РОБОТА

2.1 Обґрунтування вибору компонентів для апаратної реалізації

2.1.1 Плата BBC micro:bit як основа управління

У процесі проектування міні-робота одним із ключових рішень стало вибір апаратної платформи, яка б відповідала критеріям компактності, функціональної насиченості, енергоефективності та простоти інтеграції. Враховуючи ці вимоги, для реалізації системи керування було обрано мікроконтролерну плату BBC micro:bit.

Плата BBC micro:bit розроблена спеціально для навчання основам програмування, робототехніки та електроніки, що робить її ідеальним кандидатом для створення прототипу міні-робота. Вона має компактні розміри, що дозволяє інтегрувати її в невеликі мобільні конструкції без необхідності додаткового зменшення габаритів проектованого пристрою.

Ключовими перевагами вибору BBC micro:bit стали інтегровані сенсори (вбудований акселерометр, магнітометр і температурний сенсор дозволяють реалізовувати базову навігацію та орієнтацію без необхідності додаткових модулів), бездротова комунікація (наявність модуля BLE відкриває можливість бездротового керування роботом або обміну даними між декількома пристроями) та програмована LED-матриця 5×5, що дозволяє реалізувати візуальні індикатори стану робота без використання зовнішніх дисплеїв.

Підтримка блокового програмування в середовищі MakeCode та текстового програмування на MicroPython забезпечує легкість створення та налагодження програмного забезпечення навіть на початкових етапах розробки.

Особливо слід зазначити, що BBC micro:bit підтримує роботу з зовнішніми модулями через інтерфейси I2C, SPI та UART, що суттєво розширює її можливості при підключенні сенсорів, приводів та інших периферійних пристроїв. Така відкритість до розширення дозволяє адаптувати міні-робота до різних сценаріїв використання без потреби в зміні базової архітектури системи.

Таким чином, плата BBC micro:bit повністю відповідає поставленим технічним вимогам до системи управління міні-робота та забезпечує необхідний баланс між функціональністю, компактністю та енергоефективністю.

2.1.2 Сенсори для орієнтації та виявлення перешкод

Для забезпечення автономної навігації та ефективної взаємодії міні-робота з навколишнім середовищем було обрано набір сенсорних пристроїв, які забезпечують визначення положення, виявлення руху та ідентифікацію перешкод. Під час вибору сенсорів особлива увага приділялася критеріям компактності, енергоефективності та сумісності з платформою BBC micro:bit. У таблиці 2.1 основні характеристики обраних сенсорів, а також потенційні фактори, що можуть впливати на точність їх роботи у реальних умовах експлуатації.

Таблиця 2.1 – Сенсори для орієнтації та виявлення перешкод у міні-роботі на базі BBC micro:bit

Назва сенсора	Особливості використання	Можливі перешкоди
вбудований акселерометр (micro:bit)	реалізація функцій детекції нахилу і зіткнень без додаткових модулів	механічні вібрації, електромагнітні завади
вбудований магнітометр (micro:bit)	допомога у визначенні напрямку руху та коригуванні маршруту	зовнішні магнітні поля, металеві конструкції поблизу
ультразвуковий сенсор HC-SR04	вимірювання відстані до об'єктів на основі часу відбиття ультразвуку	акустичні шуми, м'які або дуже тонкі об'єкти
інфрачервоний сенсор TCRT5000	швидка реакція, компактність, низьке енергоспоживання	зовнішнє освітлення, відбивна поверхня

Отже, інтеграція обраних сенсорних модулів дозволяє міні-роботу на базі BBC micro:bit здійснювати базову орієнтацію, виявляти перешкоди та реагувати на зміни середовища. Використання вбудованих сенсорів плати у поєднанні з зовнішніми модулями значно розширює функціональні можливості системи без ускладнення апаратної архітектури. Водночас ідентифіковані потенційні перешкоди у роботі сенсорів слід враховувати під час проектування алгоритмів обробки даних і ухвалення рішень, що дозволить підвищити надійність та ефективність роботи міні-робота у різних середовищах.

2.1.3 Приводи для забезпечення руху робота

Організація механічного руху є однією з ключових задач у розробці міні-робота. Для реалізації функцій пересування було здійснено вибір приводів, що забезпечують необхідну мобільність при збереженні компактності конструкції та енергоефективності роботи системи. Вибір приводів здійснювався з урахуванням сумісності з платформою BBC micro:bit, вимог до розмірів, споживаної потужності та простоти керування.

Основним типом приводів для міні-робота обрано двигуни постійного струму (DC motors) малого розміру. Використання таких моторів дозволяє реалізувати ефективно переміщення мобільної платформи шляхом незалежного керування швидкістю та напрямком обертання коліс. Керування двигунами здійснюється за допомогою широтно-імпульсної модуляції (PWM), що дозволяє плавно регулювати швидкість руху робота.

Для інтеграції двигунів з BBC micro:bit використовується спеціальний драйвер моторів, наприклад TB6612FNG (рис. 2.1) або вбудований контролер у складі робототехнічних платформ типу Maqueen (рис. 2.2). Драйвер забезпечує захист мікроконтролера від перевантаження та дозволяє одночасно керувати кількома двигунами.

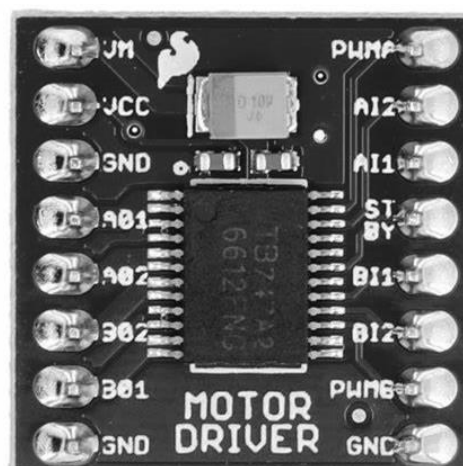


Рисунок 2.1 – Драйвер двигуна двоканальний на мікросхемі TB6612FNG [21]

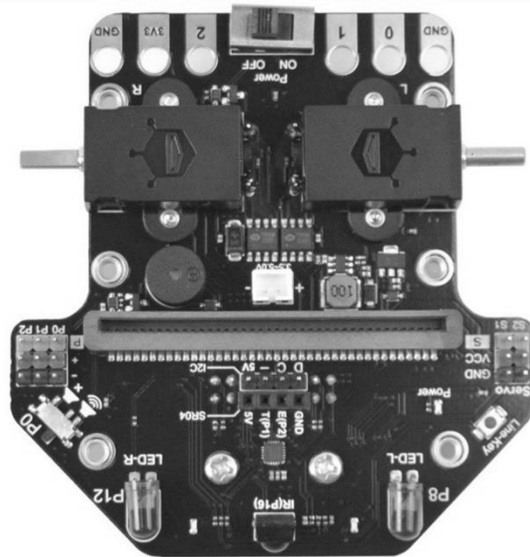


Рисунок 2.2 – Робот-програміст micro:bit [12]

Крім моторів постійного струму, для реалізації більш точного керування певними елементами конструкції передбачено використання серводвигунів типу SG90 (рис. 2.3). Ці приводи забезпечують точне позиціонування вала у заданому кутовому положенні, що є корисним для керування допоміжними механізмами, такими як маніпулятори або поворотні сенсорні модулі.



Рисунок 2.3 – Серводвигун SG90 [22]

Основними перевагами обраних приводів для міні-робота на базі BBC micro:bit є їх компактні розміри, що дозволяють інтегрувати виконавчі механізми без істотного збільшення габаритів конструкції, зберігаючи мобільність та маневреність пристрою. Завдяки мініатюрним габаритам моторів та сервоприводів можливе ефективне компонування апаратної частини навіть у межах обмеженого простору платформи.

Другою важливою перевагою є низьке енергоспоживання, що особливо критично для автономних систем, де ресурс батареї обмежений. Невисока потужність двигунів дозволяє забезпечити тривалу роботу міні-робота без необхідності частого перезарядження або заміни елементів живлення.

Крім того, обрані приводи відзначаються простою схемою управління, що суттєво зменшує обчислювальне навантаження на плату BBC micro:bit і дозволяє обійтися базовими засобами генерації широтно-імпульсних сигналів без застосування додаткових обчислювальних ресурсів. Це спрощує процес розробки та налагодження системи управління.

2.2 Схемотехнічне рішення для інтеграції компонентів

Для забезпечення ефективної взаємодії усіх апаратних модулів міні-робота на базі BBC micro:bit було розроблено схемотехнічне рішення, яке враховує особливості електричних характеристик кожного підключеного пристрою, сумісність інтерфейсів і вимоги до енергоспоживання системи.

Основною керуючою одиницею є плата BBC micro:bit, яка забезпечує обробку даних від сенсорів та формування керуючих сигналів для приводів. Комунікація з сенсорними модулями та виконавчими механізмами реалізується за допомогою цифрових і аналогових ліній введення/виведення (GPIO), а також стандартних інтерфейсів передачі даних, таких як I2C для сенсорів і PWM для управління моторами.

Ультразвуковий сенсор HC-SR04 підключається до цифрових пінів micro:bit через адаптер рівня напруги (рис. 2.4), оскільки стандартний вихідний

рівень HC-SR04 становить 5 В, що перевищує допустиме значення для входів micro:bit (3,3 В).

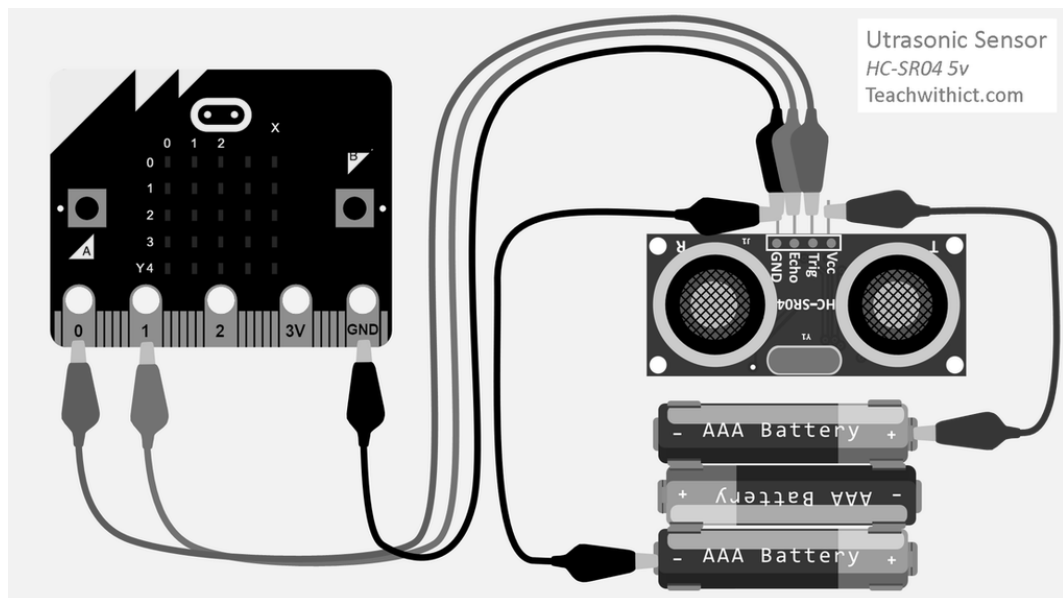


Рисунок 2.4 – Схема підключення ультразвукового датчика відстані HC-SR04 [23]

Інфрачервоні сенсори типу TCRT5000 підключаються безпосередньо до цифрових пінів, оскільки вони сумісні з рівнем напруги 3,3 В, що спрощує інтеграцію без потреби у додаткових елементах захисту.

Керування моторчиками постійного струму здійснюється через драйвер моторів TB6612FNG або аналогічний модуль, що забезпечує захист ліній micro:bit від перевантаження та дозволяє регулювати швидкість обертання моторів через сигнали широтно-імпульсної модуляції (PWM). Для кожного моторчика використовується пара сигналів: один для керування напрямком обертання, другий – для керування швидкістю.

Додатково передбачено підключення живлення до драйвера моторів і сенсорів через окремий модуль стабілізованого джерела напруги (5 В або 3,3 В залежно від типу підключених пристроїв). Це дозволяє зменшити навантаження на вбудований стабілізатор micro:bit і запобігти нестабільній роботі системи при підвищених енергетичних вимогах приводних механізмів.

На основі аналізу електричних характеристик була розроблена принципова схема інтеграції (рис. 2.5), яка включає:

- лінії живлення та «землі» для усіх модулів;
- лінії цифрового введення/виведення для сенсорів;
- PWM-сигнали для керування швидкістю моторів;
- логічні рівні сумісності для захисту мікроконтролера.
- можливість подальшого підключення додаткових модулів через резервні порти.

порти.

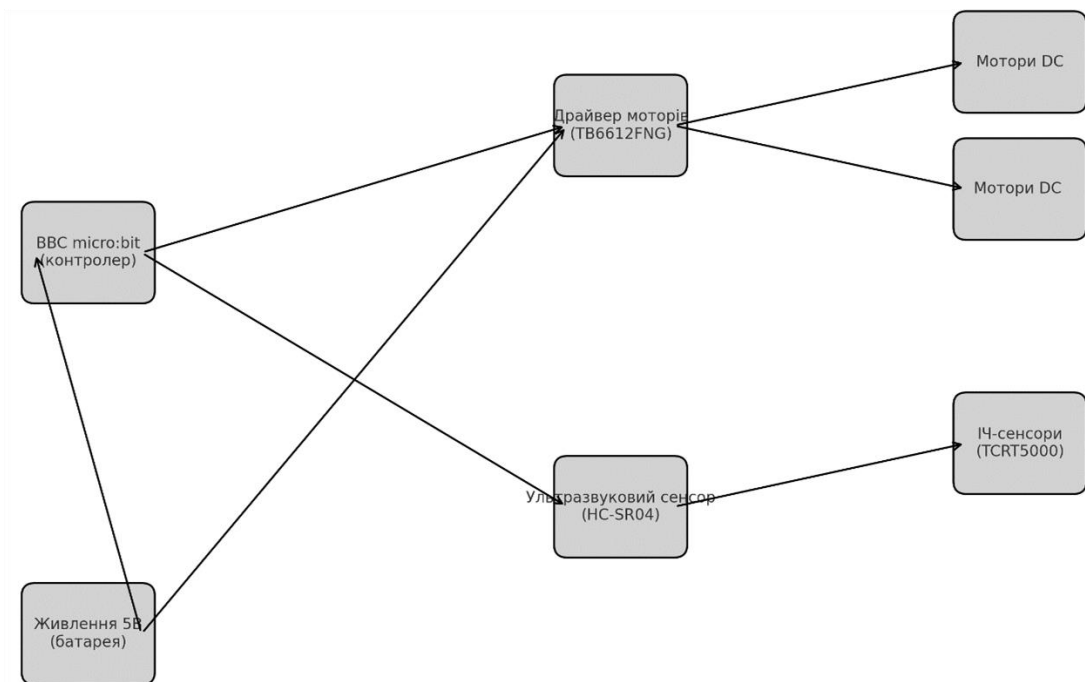


Рисунок 2.5 – Схема інтеграції міні-робота на базі BBC micro:bit

Розроблене схемотехнічне рішення забезпечує модульність системи, тобто можливість легкого розширення функціональності або заміни окремих компонентів без суттєвого перероблення всієї електричної структури. Це відповідає принципам гнучкої інженерії у проєктуванні робототехнічних систем.

Таким чином, побудована схема інтеграції забезпечує стабільну взаємодію між обраними апаратними компонентами міні-робота, гарантує безпечне підключення модулів до плати BBC micro:bit і створює основу для ефективної реалізації програмного управління системою.

2.3 Особливості програмування BBC micro:bit: середовища та інструменти

Програмування плати BBC micro:bit відіграє ключову роль у забезпеченні взаємодії між апаратними компонентами міні-робота та реалізацією алгоритмів його поведінки.

Основним середовищем програмування для BBC micro:bit є Microsoft MakeCode, яке надає можливість створювати програми у вигляді графічних блок-схем або на мові JavaScript. Використання блокового кодування особливо актуальне на етапі розробки базових алгоритмів для навчальних і прототипних робототехнічних проєктів, оскільки значно спрощує процес написання та налагодження коду. Середовище MakeCode також включає емулятор micro:bit, який дозволяє тестувати поведінку програми без фізичного підключення плати.

Для розробників, які надають перевагу текстовому програмуванню, доступна підтримка MicroPython – адаптованої версії мови Python для мікроконтролерів. Програмування на MicroPython відкриває можливості для створення більш складних логічних конструкцій, роботи з перериваннями, обробки потоків даних та реалізації асинхронних алгоритмів. При використанні MicroPython можливо більш гнучко керувати апаратними ресурсами плати, зокрема, інтерфейсами введення/виведення, PWM-сигналами та обробкою сенсорних даних.

Для полегшення процесу прошивки плати передбачено можливість програмування за допомогою простого перетягування скомпільованого файлу формату .hex у пам'ять пристрою, який розпізнається комп'ютером як звичайний USB-накопичувач. Це дозволяє швидко завантажувати нові версії програм без потреби у спеціальних драйверах чи складному налаштуванні середовища.

Серед додаткових інструментів, що підтримують розробку під BBC micro:bit, можна відзначити:

- mu-editor – легкий текстовий редактор для програмування на MicroPython [24];

- Online Python Editor для micro:bit – хмарне середовище розробки, яке дозволяє працювати без встановлення локального ПЗ;
- MakeCode Extensions – розширення для роботи з додатковими модулями, такими як ультразвукові сенсори, драйвери моторів тощо.

Для реалізації функціональності міні-робота на базі BBC micro:bit значну роль відіграє використання готових бібліотек та розширень, які дозволяють спростити розробку програмного забезпечення, прискорити інтеграцію зовнішніх модулів та зменшити обсяг низькорівневого коду. Бібліотеки надають готові інтерфейси для керування двигунами, сенсорами, світлодіодами та дисплеями, що дає змогу розробнику зосередитись на логіці взаємодії системи. Наприклад, для управління ультразвуковим сенсором HC-SR04 або драйвером TB6612FNG існують спеціалізовані програмні модулі, які дозволяють мінімізувати обсяг низькорівневого коду і сконцентруватися на логіці поведінки робота.

У таблиці 2.2 представлено найпоширеніші бібліотеки та розширення для середовищ MakeCode і MicroPython, які використовуються під час програмування компонентів типових міні-робототехнічних платформ.

Таблиця 2.2 – Приклади бібліотек та розширень для програмування BBC micro:bit [5, 12, 25]

Назва бібліотеки / розширення	Призначення	Сумісне середовище
maqueen	керування моторчиками, ультразвуковим сенсором та LED-індикацією платформи Maqueen	MakeCode
HCSR04	підключення ультразвукового сенсора HC-SR04	MakeCode
microbit_motor	керування двигунами через драйвери типу TB6612FNG	MicroPython
neopixel	керування адресними світлодіодами (RGB-LED)	MakeCode / MicroPython
microbit_i2c_lcd	підключення LCD-дисплеїв по I2C-протоколу	MicroPython

Наведений перелік бібліотек і розширень свідчить про широкий інструментарій, доступний для розробки програмного забезпечення під платформу BBC micro:bit. Завдяки сумісності з популярними середовищами MakeCode та MicroPython, більшість апаратних модулів, що використовуються у

міні-робототехніці, можуть бути швидко інтегровані без необхідності реалізації власних драйверів. Це дозволяє значно скоротити час на розробку та налагодження системи, зменшити ймовірність помилок на рівні взаємодії з обладнанням і забезпечити масштабованість проєкту.

Таким чином, гнучкість у виборі середовища програмування та наявність широкого набору інструментів забезпечують можливість ефективної розробки програмного забезпечення для міні-робота на базі BBC micro:bit з урахуванням вимог до складності проєкту та рівня підготовки розробника.

РОЗДІЛ 3

РОЗРОБКА АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МІНІ-РОБОТА

3.1 Комплектуючі міні-робота

Для початку було здійснено підготовку всіх складових міні-робота, а саме:

– комплектуючі рами (рис. 3.1);

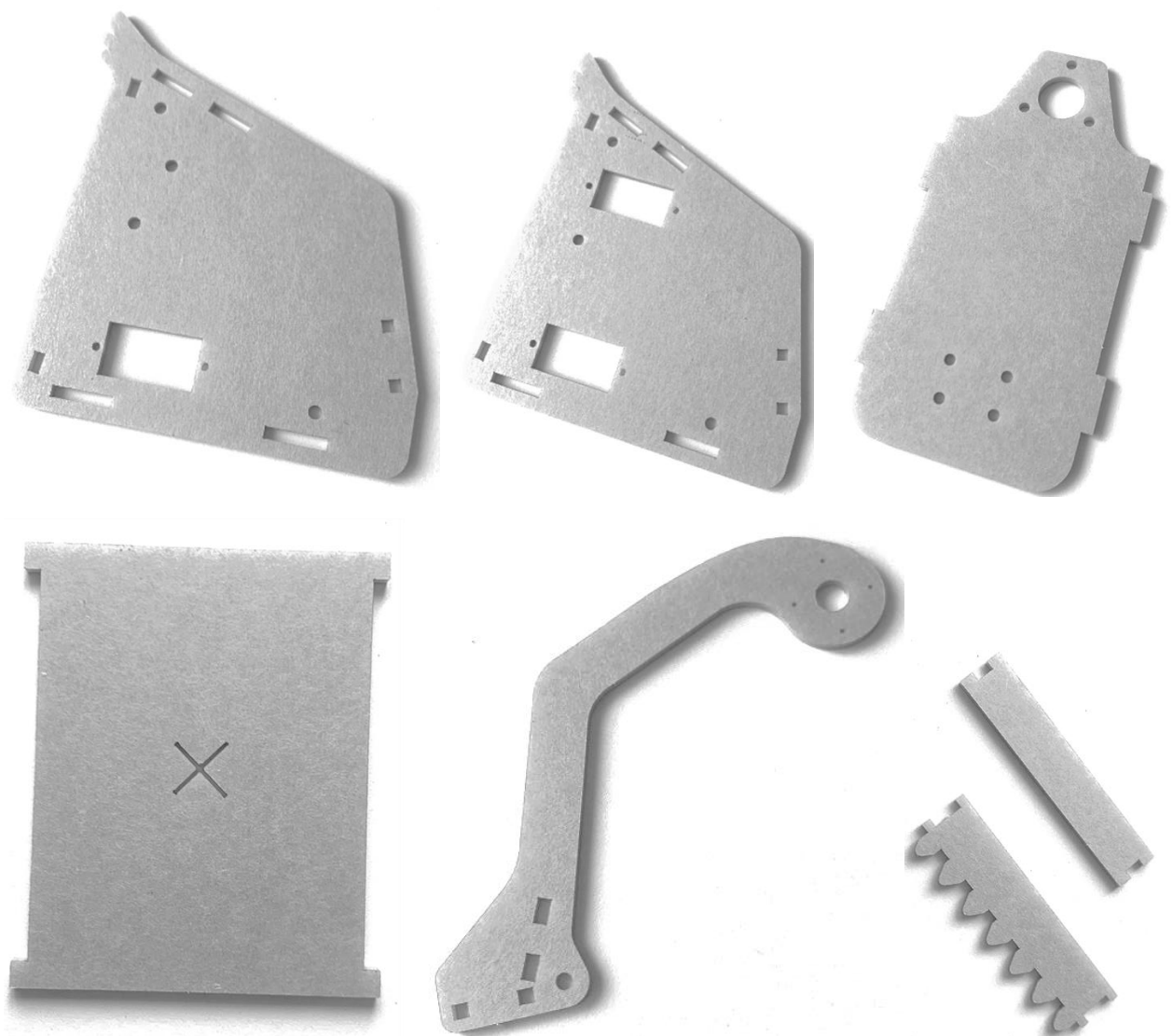


Рисунок 3.1 – Комплектуючі рами

– сервопривід (серводвигун) SG90 (рис. 3.2);



Рисунок 3.2 – Сервопривід SG90

– опорне колесо (рис. 3.3);

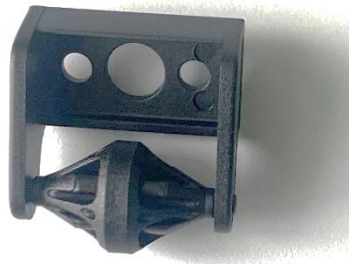


Рисунок 3.3 – Опорне колесо

– привідні колеса (рис. 3.4);



Рисунок 3.4 – Привідні колеса

– плата розширення (рис. 3.5);

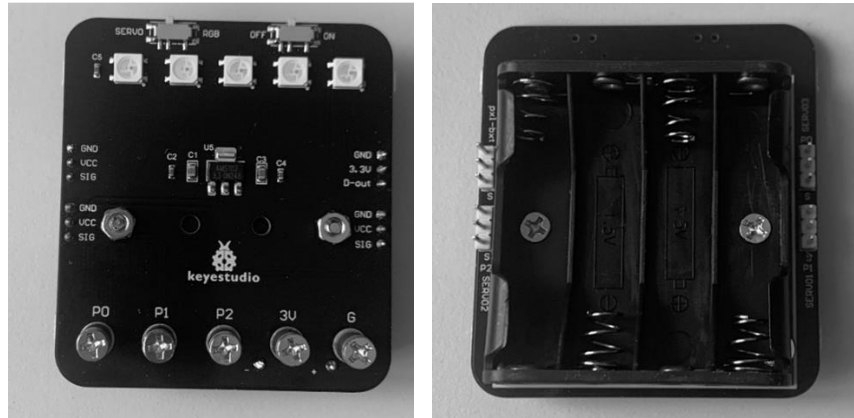


Рисунок 3.5 – Плата розширення

– плата BBC micro:bit (рис. 3.6);

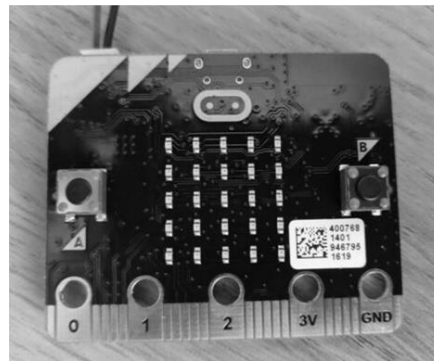
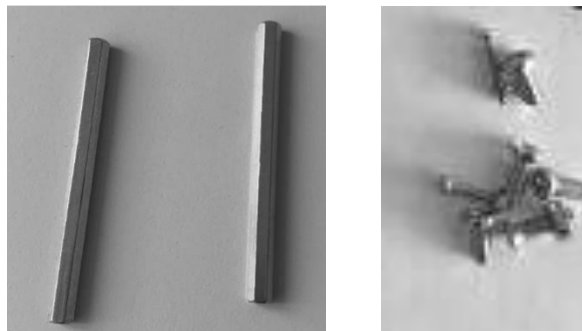


Рисунок 3.6 – Плата BBC micro:bit

– кріпильні елементи (рис. 3.7).



а)

б)

Рисунок 3.7 – Кріпильні елементи: а) двосторонні мідні стійки; б) гвинти

3.2 Складання міні-робота

Першим кроком було з'єднання комплектуючих рами, а саме бокові з нижньою основою, як представлено на рисунку 3.8.

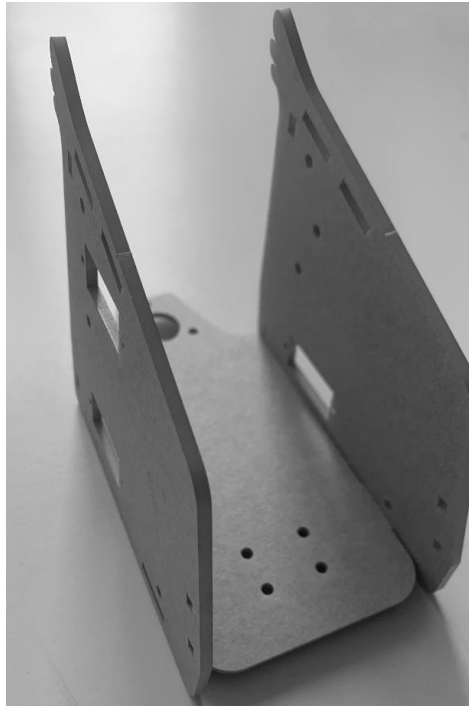
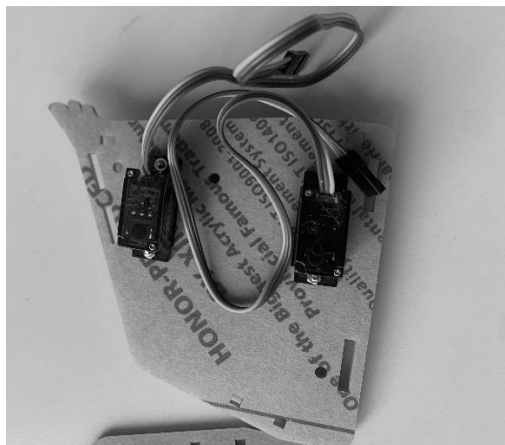
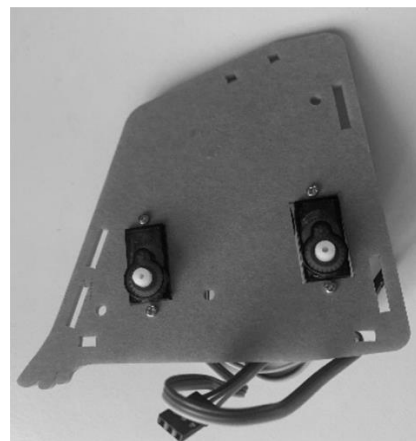


Рисунок 3.8 – З'єднання комплектуючих рами

Далі потрібно прикріпити сервоприводи, за допомогою двох гвинтів, до бокової стінки, як показано на рисунку 3.9.



а)



б)

Рисунок 3.9 – З'єднання сервоприводів: а) вигляд зсередини; б) вигляд ззовні

Наступним кроком було кріплення коліс (рис. 3.10).

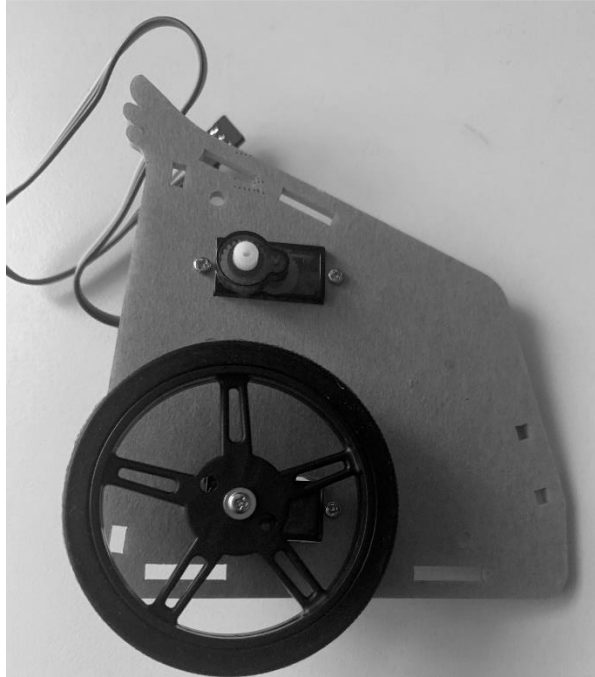


Рисунок 3.10 – Кріплення коліс до бокових стінок

Далі сполучили обидві бокові стінки з колесами за допомогою двосторонньої мідної стійки, як представлено на рисунку 3.11.



Рисунок 3.11 – З'єднання бокових стінок за допомогою мідної стійки

Наступним кроком буде приєднання опорного колеса до нижньої платформи за допомогою гвинтів та гайок (рис. 3.12).



Рисунок 3.12 – З'єднання опорного колеса

Після з'єднання опорного колеса складається ковш для різного призначення міні-робота (рис. 3.13).



Рисунок 3.13 – Ковш міні-робота.

Для руху ковша прикручується хрестовина щоб було з'єднання з сервоприводом, як представлено на рисунку 3.14, та приєднується до попередньо зробленого міні-робота (рис. 3.15).



Рисунок 3.14 – Хрестовина для з'єднання із сервоприводом

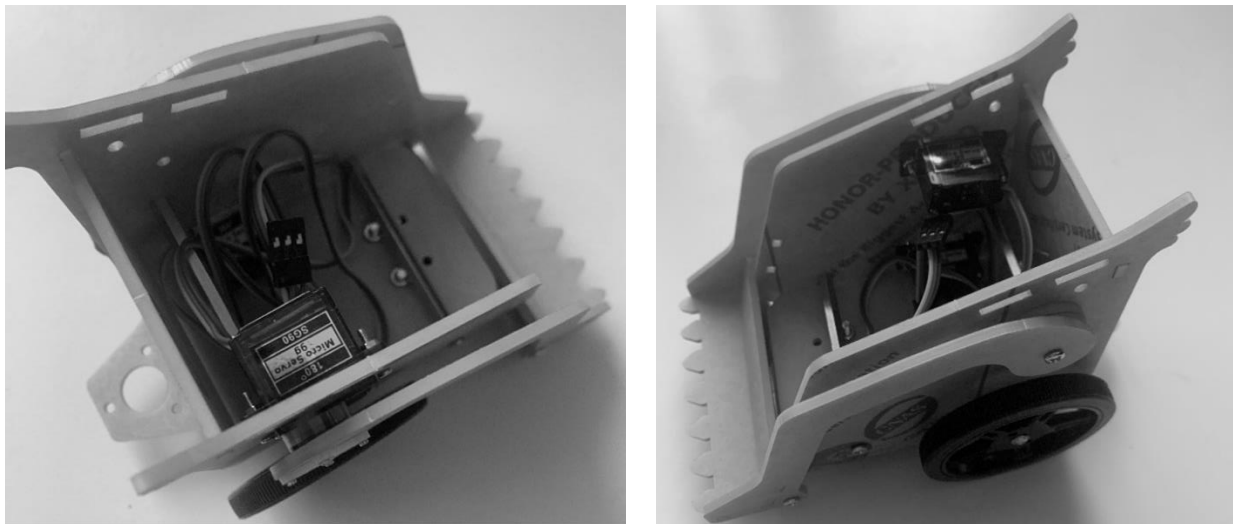


Рисунок 3.15 – Приєднання ковша до міні-робота

Наступним кроком було з'єднання плати розширення з платою ВВС micro:bit за допомогою гвинтів, як представлено на рисунку 3.16.

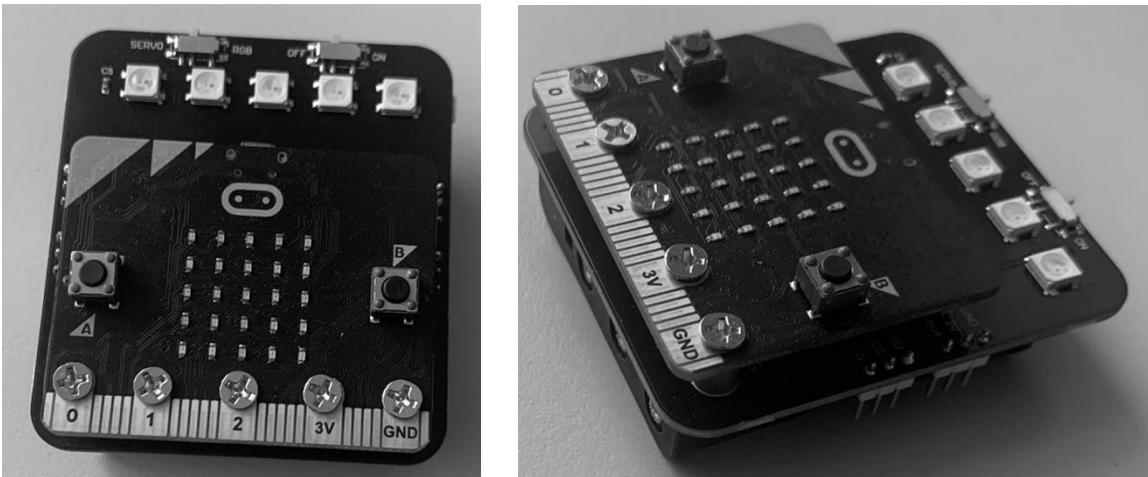


Рисунок 3.16 – З'єднання плати розширення з платою BBC micro:bit

Далі встановлюється вставна частина плати керування (рис. 3.17) для приєднання плати BBC micro:bit до корпусу міні-робота (рис. 3.18).

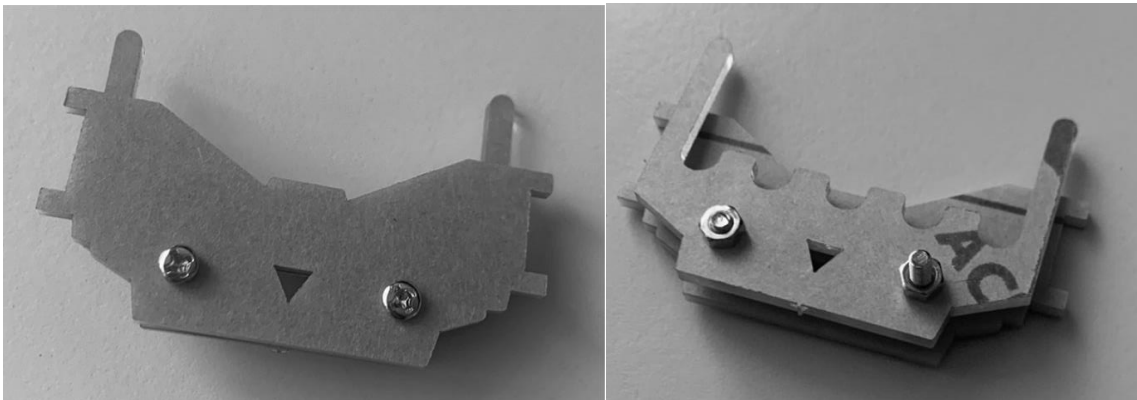


Рисунок 3.17 – вставна частина плати керування

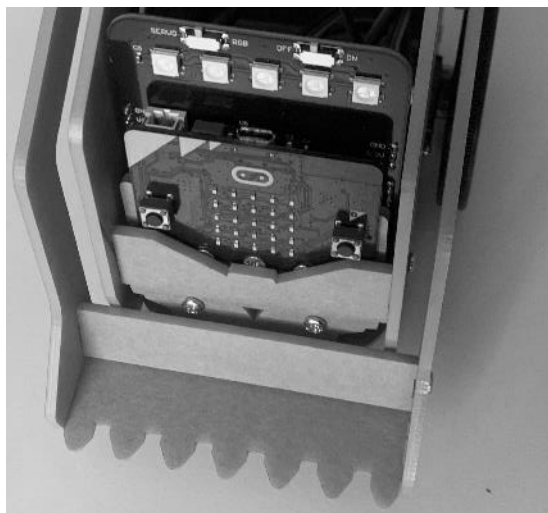


Рисунок 3.18 – Приєднання плати BBC micro:bit до корпусу міні-робота

Для керування за допомогою BBC micro:bit з'єднано сервоприводи до плати розширення, згідно рисунку 3.19.

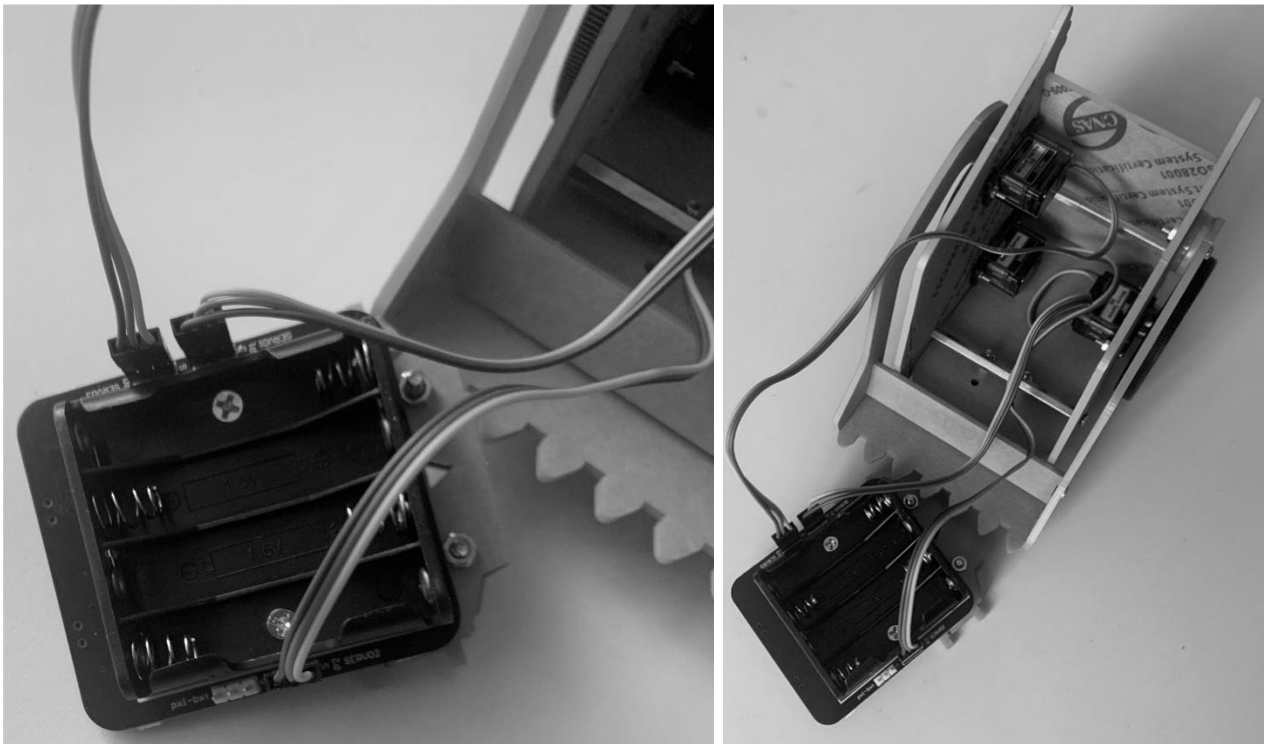


Рисунок 3.19 – З'єднання сервоприводів до плати розширення

Далі з'єнується верхня кришка та тримач для фломастера ззаду міні-робота, як представлено на рисунку 3.20.

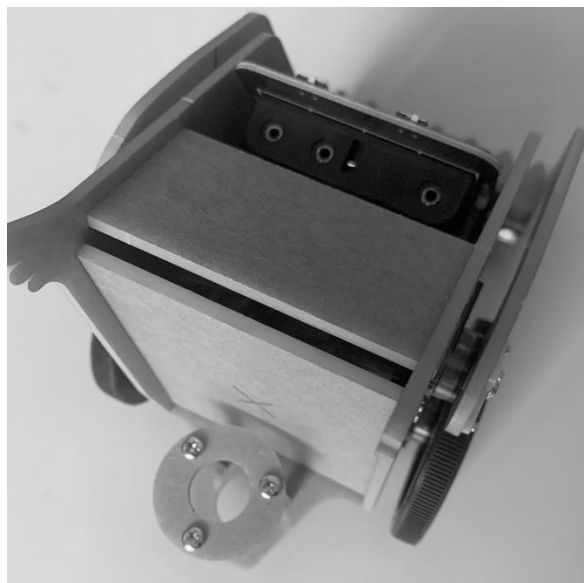


Рисунок 3.20 – З'єднання верхньої кришки та тримач для фломастера

Складений міні-робота на основі плати BBC micro:bit представлено на рисунку 3.21.



Рисунок 3.21 – Складений міні-робота на основі плати BBC micro:bit

3.3 Програмування плати BBC micro:bit для дистанційного керування міні-роботом

Для розробки було взято операційну систему Windows. Підключаємо micro:bit до комп'ютера за допомогою USB-кабелю (рис. 3.22).

Після підключення до комп'ютера, червоний світлодіод micro:bit V2 засвітиться.

Далі на персональному комп'ютері буде встановлений дисковод MICROBIT, як показано на рисунку 3.23.



Рисунок 3.22 – Підключення micro:bit до комп'ютера за допомогою USB-кабелю

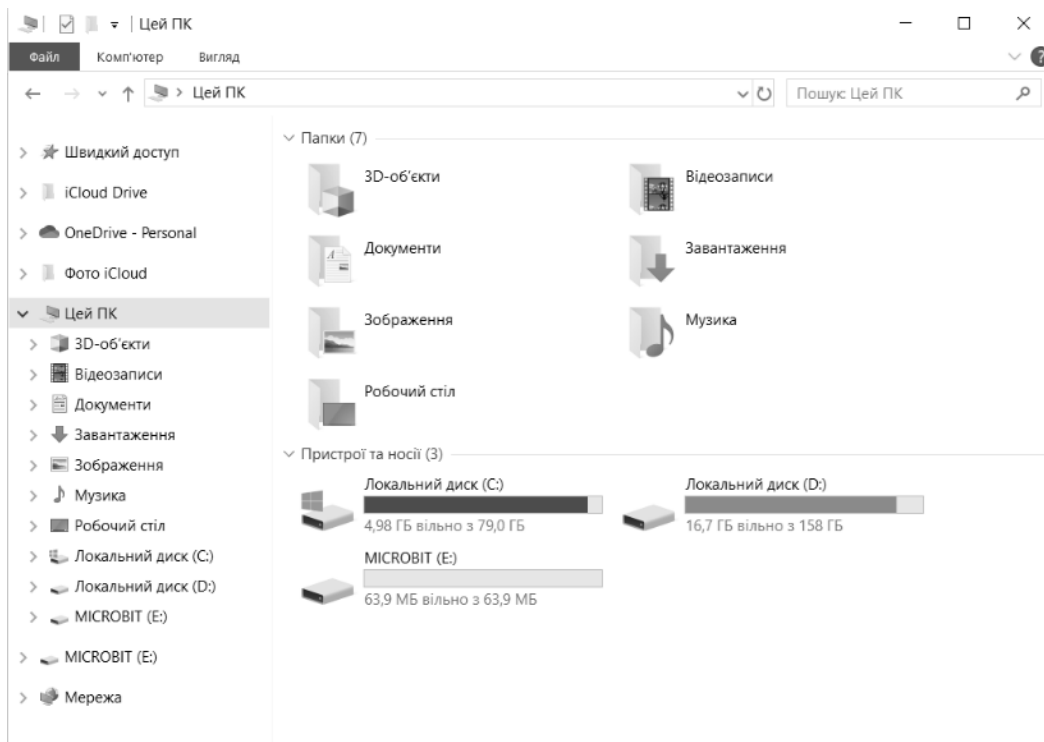


Рисунок 3.23 – Встановлення дисководу MICROBIT

Після цього потрібно перейти на сайт <https://makecode.microbit.org> та натиснути новий створити новий проєкт (рис. 3.24).

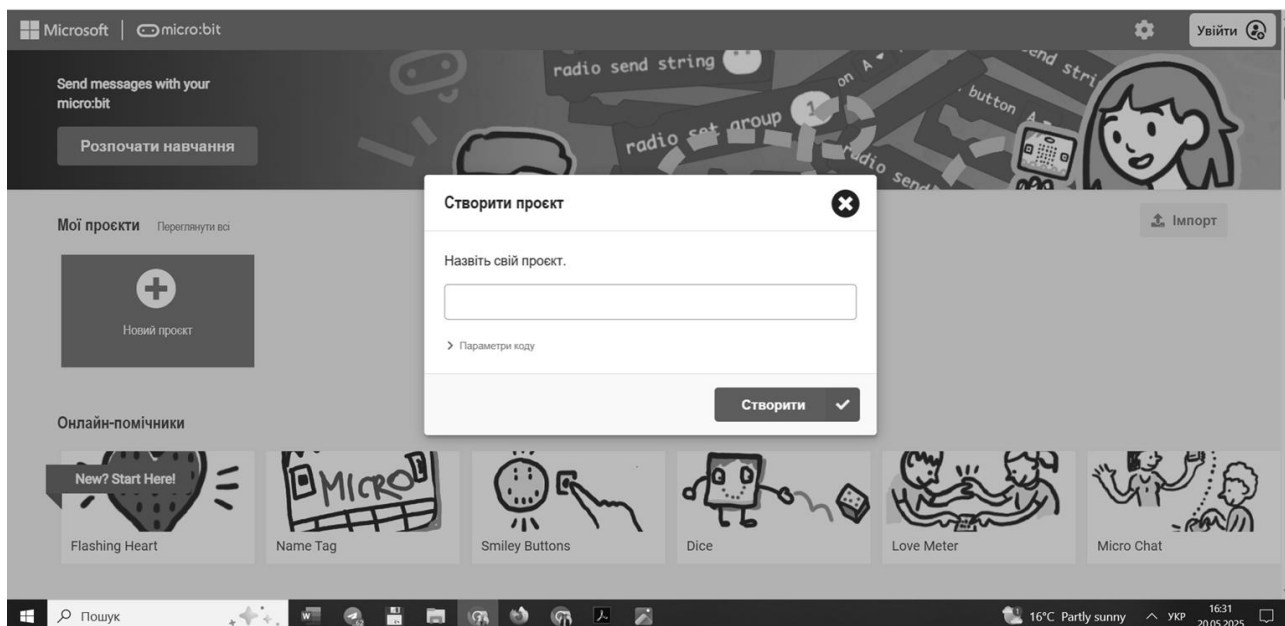


Рисунок 3.24 – Створення нового проєкту

Через редактор MakeCode (рис. 3.25) можна просто потрібно перетягнути блоки з області блоків в область редагування коду для програмування або писати код на JavaScript чи Python (рис. 3.26).

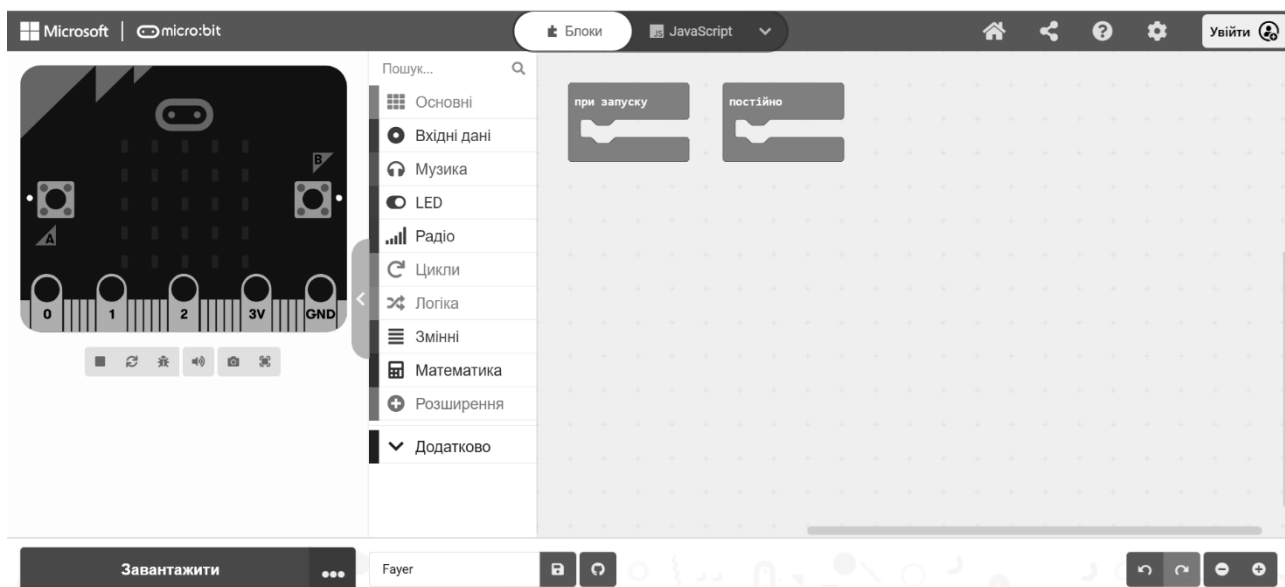


Рисунок 3.25 – Редактор MakeCode

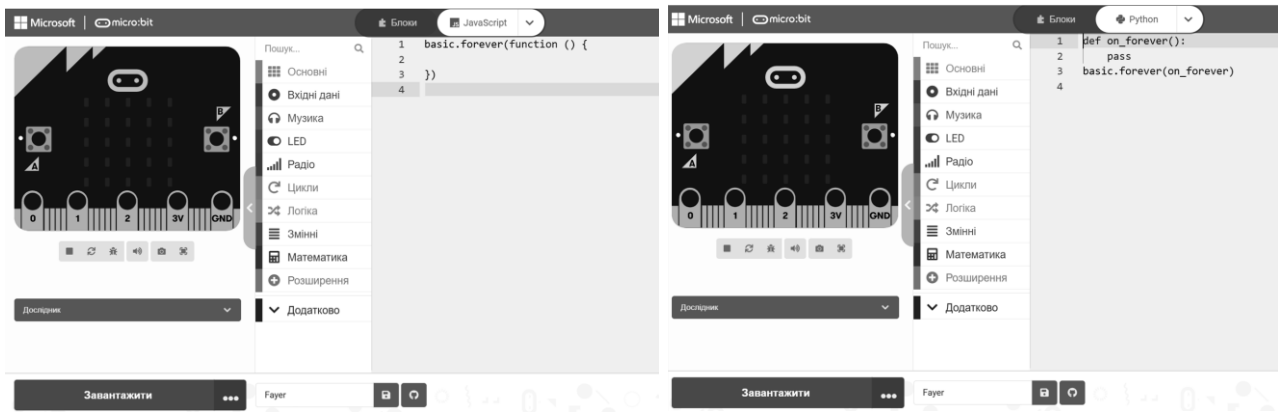


Рисунок 3.26 – Редактор MakeCode в режимі програмування JavaScript та Python

Після написання коду в тестовому режимі можна запустити та побачити чи все працює без помилок, як на рисунку 3.27. А сам власне код представлено в додатку А та Б.

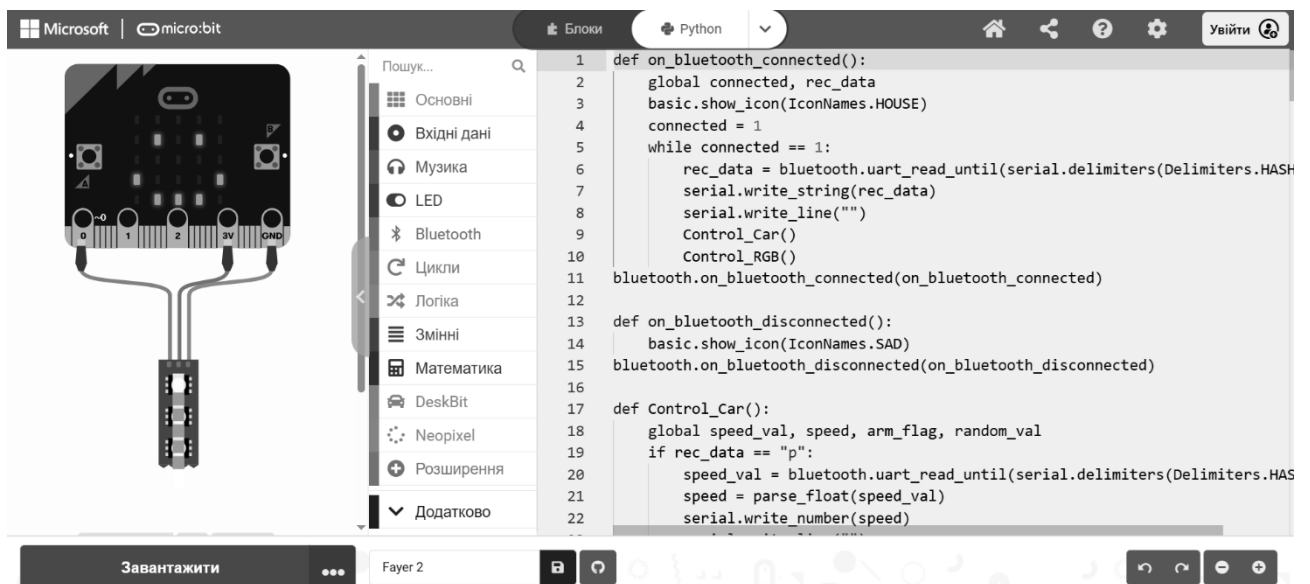


Рисунок 3.27 – Написаний код в додатку makescode

Код можна завантажити безпосередньо на плату, якщо натиснути значок «Завантажити» у застосунку makecode (рис. 3.28).



Рисунок 3.28 – Процес завантаження коду

Жовтий індикатор блиматиме під час передачі «hex»-коду на плату ВВС micro:bit та світлитиметься постійно після копіювання файлу.

ВИСНОВКИ

Отримані результати роботи дозволяють сформулювати висновки щодо технічної доцільності, переваг та обмежень використання обраної платформи для реалізації поставлених завдань.

У процесі порівняльного аналізу сучасних апаратних рішень, встановлено, що платформа BBC micro:bit є оптимальною для створення міні-робота, яка поєднує низьку вартість, компактність, підтримку різних мов програмування та наявність вбудованих сенсорів. Порівняно з іншими платформами, BBC micro:bit забезпечує достатню функціональність для реалізації базових навігаційних і керуючих функцій без потреби у складному периферійному обладнанні.

Детально проаналізовано технічні можливості та характеристики BBC micro:bit. Визначено, що мікроконтролер здатен обробляти сигнали від зовнішніх сенсорів, керувати приводами та забезпечувати бездротову взаємодію, що повністю відповідає вимогам до платформи для створення мобільного міні-робота.

Підібрано апаратні компоненти з урахуванням енергоспоживання, сумісності та функціональних можливостей для створення міні-робота на базі плати BBC micro:bit.

Здійснено складання міні-робота на основі підібраних компонентів. Створено код в середовищі MakeCode для керування рухом та взаємодії з користувачем через бездротовий інтерфейс. Результати тестування підтвердили роботоздатність міні-робота, що виконує рух за командами та демонструє базову навігацію в контрольованому середовищі. Це доводить ефективність реалізованого апаратно-програмного рішення.

У результаті виконання кваліфікаційної роботи бакалавра було успішно реалізовано міні-робота на базі плати BBC micro:bit, що підтвердило доцільність використання цієї платформи для створення експериментальних робототехнічних систем. Проведений аналіз сучасних апаратних рішень, характеристик micro:bit, дозволив обґрунтовано сформулювати склад пристрою.

Зібраний прототип продемонстрував працездатність у реальних умовах, стабільну взаємодію між компонентами та можливість дистанційного керування, що свідчить про досягнення поставленої мети та практичну ефективність розробки.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Arduino: офіційний сайт з документацією та прикладами використання платформи. *Arduino*. URL: <https://www.arduino.cc/> (дата звернення: 25.01.2025).
2. Build an Autonomous Arduino Robot with Bump Sensors. *Science Buddies*. URL: https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p034/robotics/arduino-robot-bump-sensors (дата звернення: 25.01.2025).
3. Raspberry Pi: офіційний сайт з інформацією про моделі, технічні характеристики та проекти. *Raspberry Pi Foundation*. URL: <https://www.raspberrypi.org/> (дата звернення: 25.01.2025).
4. Somarathna R. N. Path Planning and Obstacle Avoidance Scheme for Autonomous Robots using Raspberry Pi. *arXiv*. URL: <https://arxiv.org/pdf/2012.10863> (дата звернення: 25.01.2025).
5. BBC micro:bit: офіційний сайт з ресурсами для навчання та розробки проєктів. *Освітня фундація Micro:bit*. URL: <https://microbit.org/> (дата звернення: 25.01.2025).
6. K. Hartley, E. Rubegni, L. Underwood, J. Finney, Th.s Ball, S.Hodges, P. De Halleux, J. Devine, E. Anderson, M. Moskal. Meet MicroCode: a Live and Portable Programming Tool for the BBC micro:bit. *IDC '24: Proceedings of the 23rd Annual ACM Interaction Design and Children Conference*. Delft, Netherlands, 2024. P. 355-370.
7. ESP32: документація та ресурси для розробників. *Espressif*. URL: <https://www.espressif.com/en/products/socs/esp32> (дата звернення: 25.01.2025).
8. T. Huang, R. Xiong. Cost-Effective Robotic Handwriting System with AI Integration. *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. Холтсвілл, Нью-Йорк, США, 2024. URL: <https://arxiv.org/pdf/2501.06783> (дата звернення: 25.01.2025).

9. Tinkercad Circuits: онлайн-платформа для моделювання електронних схем. *Autodesk*. URL: <https://www.tinkercad.com/circuits> (дата звернення: 25.01.2025).

10. mBot: робототехнічний набір на основі Arduino для навчання програмуванню. *Makeblock*. URL: <https://www.makeblock.com/steam-kits/mbot> (дата звернення: 25.01.2025).

11. PiCar-S: робототехнічний набір на базі Raspberry Pi для вивчення автономного водіння. *SunFounder*. URL: <https://www.sunfounder.com/products/picar-s> (дата звернення: 25.01.2025).

12. Maqueen Robot: інформація про робототехнічний набір на базі micro:bit. *DFRobot*. URL: <https://www.dfrobot.com/product-1783.html> (дата звернення: 25.01.2025).

13. Nakayiza Hellen, Ggaliwango Marvin. Explainable AI for Safe Water Evaluation for Public Health in Urban Settings. *International Conference on Innovations in Science, Engineering and Technology (ICISSET)*. Chittagong, Bangladesh, 2022. URL: <https://ieeexplore.ieee.org/document/9775912> (дата звернення: 15.01.2025).

14. Micro:bit Line Following Robot. *Autodesk*. URL: <https://www.instructables.com/Microbit-Line-Following-Robot/> (дата звернення: 05.02.2025).

15. Distance Sensing With the Micro:bit and Sonar (HC-SR04 Module). *Autodesk*. URL: <https://www.instructables.com/Distance-Sensing-With-the-Microbit-and-Sonar-HC-SR/> (дата звернення: 05.02.2025).

16. Use MakeCode to Program the micro:bit with a TCS34725 Colour Sensor. *Smalldevices*. URL: <https://smalldevices.com.au/blogs/resources/use-makecode-to-program-the-micro-bit-with-a-tcs34735-colour-sensor> (дата звернення: 05.02.2025).

17. R. Duban, Yu. Yurko. Using micro:bit for building distributed sensor networks in robotic systems. *Challenges and Issues of Modern Science*. Vol. 3. 2024. URL: <https://cims.fti.dp.ua/j/article/view/228/204> (дата звернення: 05.02.2025).

18. Lund H. H., Exner M., Jensen N.E., Leggieri M., Outzen M., Ravn-Haren G., Sehested M., Væring A., Andersen R. GrowBot: An Educational Robotic System for Growing Food. *Applied Sciences*. Vol. 12 (11). 2022. URL: https://www.researchgate.net/publication/360955254_GrowBot_An_Educational_Robotic_System_for_Growing_Food (дата звернення: 11.02.2025).

19. How do I connect microbit with BLE and listen for button press events? *Stack Exchange*. URL: <https://stackoverflow.com/questions/63750053/how-do-i-connect-microbit-with-ble-and-listen-for-button-press-events> (дата звернення: 20.02.2025).

20. Romilly Cocking. I2C and SPI on the micro:bit; additions to the babelboard range. *RAREblog. Raspberry Pi, Arduino, Robotics and Electronics*. URL: <https://blog.rareschool.com/2020/02/i2c-and-spi-on-microbit-additions-to.html> (дата звернення: 20.02.2025).

21. Драйвер TB6612FNG. *Arduino* URL: <https://arduino.ua/prod2377-draiver-dvigatelei-dvyhkanalnii-na-tb6612fng> (дата звернення: 17.02.2025).

22. Серводвигун SG90. *Arduino* URL: <https://arduino.ua/prod416-servoprivod-sg90-2kg> (дата звернення: 17.02.2025).

23. Базовий підручник BBC micro:bit 07-micro:bit та ультразвуковий датчик HC-SR04. *Arduino та micro:bit makers*. URL: <http://t.zoukankan.com/microcampus-p-9747633.html> (дата звернення: 13.03.2025).

24. Code with Mu: a simple Python editor for beginner programmers. *Nicholas H.Tollervey*. URL: <https://codewith.mu/> (дата звернення: 11.04.2025).

25. Версія мови Python для мікроконтролерів. *Освітня фундація Micro:bit* URL: <https://python.microbit.org/> (дата звернення: 11.04.2025).

ДОДАТКИ

Додаток А

Код програми в Python

```

def on_bluetooth_connected():
    global connected, rec_data
    basic.show_icon(IconNames.HOUSE)
    connected = 1
    while connected == 1:
        rec_data = bluetooth.uart_read_until(serial.delimiters(Delimiters.HASH))
        serial.write_string(rec_data)
        serial.write_line("")
        Control_Car()
        Control_RGB()
bluetooth.on_bluetooth_connected(on_bluetooth_connected)

def on_bluetooth_disconnected():
    basic.show_icon(IconNames.SAD)
bluetooth.on_bluetooth_disconnected(on_bluetooth_disconnected)

def Control_Car():
    global speed_val, speed, arm_flag, random_val
    if rec_data == "p":
        speed_val = bluetooth.uart_read_until(serial.delimiters(Delimiters.HASH))
        speed = parse_float(speed_val)
        serial.write_number(speed)
        serial.write_line("")
    if rec_data == "F":
        DeskBit.run(DIR.RUN_FORWARD, speed)
    elif rec_data == "L":
        DeskBit.run(DIR.LEFT_ROTATION, speed)
    elif rec_data == "R":
        DeskBit.run(DIR.RIGHT_ROTATION, speed)
    elif rec_data == "B":
        DeskBit.run(DIR.RUN_BACK, speed)
    elif rec_data == "Q":
        DeskBit.run2(L_R.LEFT, F_B.FORWARD, 50)
        DeskBit.run2(L_R.RIGHT, F_B.FORWARD, 100)
    elif rec_data == "E":
        DeskBit.run2(L_R.LEFT, F_B.FORWARD, 100)
        DeskBit.run2(L_R.RIGHT, F_B.FORWARD, 50)
    elif rec_data == "S":
        DeskBit.stop()
    elif rec_data == "o":
        basic.show_icon(IconNames.HOUSE)
    elif rec_data == "u":
        arm_flag = 1
    elif rec_data == "d":
        arm_flag = 2
    elif rec_data == "m":
        arm_flag = 0
    elif rec_data == "1":
        DeskBit.shovel(90)
        basic.pause(200)
        DeskBit.shovel(30)

```

```

        basic.pause(200)
        DeskBit.shovel(90)
        basic.pause(200)
        DeskBit.shovel(0)
        basic.pause(200)
    elif rec_data == "2":
        DeskBit.shovel(120)
        basic.pause(200)
        basic.show_icon(IconNames.HEART)
        basic.pause(1000)
        DeskBit.shovel(0)
        basic.pause(200)
        basic.show_icon(IconNames.HOUSE)
    elif rec_data == "3":
        DeskBit.run2(L_R.LEFT, F_B.FORWARD, 0)
        DeskBit.run2(L_R.RIGHT, F_B.FORWARD, 50)
    elif rec_data == "4":
        DeskBit.run2(L_R.LEFT, F_B.FORWARD, 50)
        DeskBit.run2(L_R.RIGHT, F_B.FORWARD, 0)
    elif rec_data == "5":
        DeskBit.run(DIR.LEFT_ROTATION, 100)
        basic.pause(1000)
        random_val = randint(1, 6)
        basic.show_number(random_val)
    elif rec_data == "6":
        for index in range(5):
            strip.set_pixel_color(index, neopixel.colors(NeoPixelColors.BLUE))
            strip.show()
            basic.pause(500)
        basic.pause(200)
        strip.clear()
        strip.show()
    elif rec_data == "n":
        DeskBit.stop()
        strip.clear()
        strip.show()
def Control_RGB():
    global RGB_val, RGB_bright, RGB_brightness
    if rec_data == "c":
        RGB_val += 1
        if RGB_val >= 9:
            RGB_val = 9
    elif rec_data == "b":
        RGB_val += -1
        if RGB_val <= 1:
            RGB_val = 1
    elif rec_data == "s":
        strip.show_color(neopixel.colors(NeoPixelColors.BLACK))
        RGB_val = 0
    elif rec_data == "g":
        RGB_bright = bluetooth.uart_read_until(serial.delimiters(Delimiters.HASH))
        RGB_brightness = parse_float(RGB_bright)
        serial.write_number(RGB_brightness)
        serial.write_line("")
        strip.set_brightness(RGB_brightness)
        strip.show()
    if RGB_val == 1:

```

```

        strip.show_color(neopixel.colors(NeoPixelColors.RED))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 2:
        strip.show_color(neopixel.colors(NeoPixelColors.ORANGE))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 3:
        strip.show_color(neopixel.colors(NeoPixelColors.YELLOW))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 4:
        strip.show_color(neopixel.colors(NeoPixelColors.GREEN))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 5:
        strip.show_color(neopixel.colors(NeoPixelColors.BLUE))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 6:
        strip.show_color(neopixel.colors(NeoPixelColors.INDIGO))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 7:
        strip.show_color(neopixel.colors(NeoPixelColors.VIOLET))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 8:
        strip.show_color(neopixel.colors(NeoPixelColors.PURPLE))
        strip.set_brightness(RGB_brightness)
        strip.show()
    elif RGB_val == 9:
        strip.show_color(neopixel.colors(NeoPixelColors.WHITE))
        strip.set_brightness(RGB_brightness)
        strip.show()
def Control_Arm():
    global arm_angle
    if arm_flag == 1:
        arm_angle += 1
        serial.write_string("" + str((arm_angle)))
        serial.write_line("")
        if arm_angle >= 180:
            arm_angle = 180
        DeskBit.shovel(arm_angle)
    if arm_flag == 2:
        arm_angle += -1
        serial.write_string("" + str((arm_angle)))
        serial.write_line("")
        if arm_angle <= 0:
            arm_angle = 0
        DeskBit.shovel(arm_angle)
RGB_bright = ""
random_val = 0
speed_val = ""
rec_data = ""
connected = 0
RGB_brightness = 0

```

```
speed = 0
RGB_val = 0
arm_flag = 0
arm_angle = 0
strip: neopixel.Strip = None
serial.redirect_to_usb()
strip = neopixel.create(DigitalPin.P0, 5, NeoPixelMode.RGB)
DeskBit.shovel(0)
basic.show_icon(IconNames.HAPPY)
arm_angle = 0
arm_flag = 0
RGB_val = 0
speed = 100
RGB_brightness = 150
strip.clear()
strip.show()

def on_forever():
    Control_Arm()
basic.forever(on_forever)
```

Додаток Б

Код програми в JavaScript

```

bluetooth.onBluetoothConnected(function () {
    basic.showIcon(IconNames.House)
    connected = 1
    while (connected == 1) {
        rec_data = bluetooth.uartReadUntil(serial.delimiters(Delimiters.Hash))
        serial.writeString(rec_data)
        serial.writeLine("")
        Control_Car()
        Control_RGB()
    }
})
bluetooth.onBluetoothDisconnected(function () {
    basic.showIcon(IconNames.Sad)
})
function Control_Car () {
    if (rec_data == "p") {
        speed_val = bluetooth.uartReadUntil(serial.delimiters(Delimiters.Hash))
        speed = parseFloat(speed_val)
        serial.writeNumber(speed)
        serial.writeLine("")
    }
    if (rec_data == "F") {
        DeskBit.Run(DIR.runForward, speed)
    } else if (rec_data == "L") {
        DeskBit.Run(DIR.leftRotation, speed)
    } else if (rec_data == "R") {
        DeskBit.Run(DIR.rightRotation, speed)
    } else if (rec_data == "B") {
        DeskBit.Run(DIR.runBack, speed)
    } else if (rec_data == "Q") {
        DeskBit.Run2(L_R.left, F_B.forward, 50)
        DeskBit.Run2(L_R.right, F_B.forward, 100)
    } else if (rec_data == "E") {
        DeskBit.Run2(L_R.left, F_B.forward, 100)
        DeskBit.Run2(L_R.right, F_B.forward, 50)
    } else if (rec_data == "S") {
        DeskBit.Stop()
    } else if (rec_data == "o") {
        basic.showIcon(IconNames.House)
    } else if (rec_data == "u") {
        arm_flag = 1
    } else if (rec_data == "d") {
        arm_flag = 2
    } else if (rec_data == "m") {
        arm_flag = 0
    } else if (rec_data == "1") {
        DeskBit.Shovel(90)
        basic.pause(200)
        DeskBit.Shovel(30)
        basic.pause(200)
        DeskBit.Shovel(90)
    }
}

```

```

        basic.pause(200)
        DeskBit.Shovel(0)
        basic.pause(200)
    } else if (rec_data == "2") {
        DeskBit.Shovel(120)
        basic.pause(200)
        basic.showIcon(IconNames.Heart)
        basic.pause(1000)
        DeskBit.Shovel(0)
        basic.pause(200)
        basic.showIcon(IconNames.House)
    } else if (rec_data == "3") {
        DeskBit.Run2(L_R.left, F_B.forward, 0)
        DeskBit.Run2(L_R.right, F_B.forward, 50)
    } else if (rec_data == "4") {
        DeskBit.Run2(L_R.left, F_B.forward, 50)
        DeskBit.Run2(L_R.right, F_B.forward, 0)
    } else if (rec_data == "5") {
        DeskBit.Run(DIR.leftRotation, 100)
        basic.pause(1000)
        random_val = randint(1, 6)
        basic.showNumber(random_val)
    } else if (rec_data == "6") {
        for (let index = 0; index <= 4; index++) {
            strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Blue))
            strip.show()
            basic.pause(500)
        }
        basic.pause(200)
        strip.clear()
        strip.show()
    } else if (rec_data == "n") {
        DeskBit.Stop()
        strip.clear()
        strip.show()
    }
}

function Control_RGB () {
    if (rec_data == "c") {
        RGB_val += 1
        if (RGB_val >= 9) {
            RGB_val = 9
        }
    } else if (rec_data == "b") {
        RGB_val += -1
        if (RGB_val <= 1) {
            RGB_val = 1
        }
    } else if (rec_data == "s") {
        strip.showColor(neopixel.colors(NeoPixelColors.Black))
        RGB_val = 0
    } else if (rec_data == "g") {
        RGB_bright = bluetooth.uartReadUntil(serial.delimiters(Delimiters.Hash))
        RGB_brightness = parseFloat(RGB_bright)
        serial.writeNumber(RGB_brightness)
        serial.writeLine("")
        strip.setBrightness(RGB_brightness)
    }
}

```

```

        strip.show()
    }
    if (RGB_val == 1) {
        strip.showColor(neopixel.colors(NeoPixelColors.Red))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 2) {
        strip.showColor(neopixel.colors(NeoPixelColors.Orange))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 3) {
        strip.showColor(neopixel.colors(NeoPixelColors.Yellow))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 4) {
        strip.showColor(neopixel.colors(NeoPixelColors.Green))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 5) {
        strip.showColor(neopixel.colors(NeoPixelColors.Blue))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 6) {
        strip.showColor(neopixel.colors(NeoPixelColors.Indigo))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 7) {
        strip.showColor(neopixel.colors(NeoPixelColors.Violet))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 8) {
        strip.showColor(neopixel.colors(NeoPixelColors.Purple))
        strip.setBrightness(RGB_brightness)
        strip.show()
    } else if (RGB_val == 9) {
        strip.showColor(neopixel.colors(NeoPixelColors.White))
        strip.setBrightness(RGB_brightness)
        strip.show()
    }
}
function Control_Arm () {
    if (arm_flag == 1) {
        arm_angle += 1
        serial.writeString("" + (arm_angle))
        serial.writeLine("")
        if (arm_angle >= 180) {
            arm_angle = 180
        }
        DeskBit.Shovel(arm_angle)
    }
    if (arm_flag == 2) {
        arm_angle += -1
        serial.writeString("" + (arm_angle))
        serial.writeLine("")
        if (arm_angle <= 0) {
            arm_angle = 0
        }
    }
}

```

```
        DeskBit.Shovel(arm_angle)
    }
}
let RGB_bright = ""
let random_val = 0
let speed_val = ""
let rec_data = ""
let connected = 0
let RGB_brightness = 0
let speed = 0
let RGB_val = 0
let arm_flag = 0
let arm_angle = 0
let strip: neopixel.Strip = null
serial.redirectToUSB()
strip = neopixel.create(DigitalPin.P0, 5, NeoPixelMode.RGB)
DeskBit.Shovel(0)
basic.showIcon(IconNames.Happy)
arm_angle = 0
arm_flag = 0
RGB_val = 0
speed = 100
RGB_brightness = 150
strip.clear()
strip.show()
basic.forever(function () {
    Control_Arm()
})
```