БАКАЛАВРСЬКА РОБОТА

ПМІ.БР-60.00.00.000 ПЗ

Група ПМІ-21-1

Дрегало Олексій

2025

Івано-Франківський національний технічний університет нафти і газу
Інститут інженерної механіки та робототехніки
Кафедра: комп'ютеризованого машинобудування

Дрегало Олексій Юрійович

УДК 62-5.007.5

БАКАЛАВРСЬКА РОБОТА

Розробка моделі марсоходу
(назва роботи)

Інженерія мехатронних систем
(назва освітньої програми)

131 – Прикладна механіка
(шифр і назва спеціальності)

О. Ю. Дрегало
(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Панчук В. Г., професор ,зав. кафедри КМВ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту
Завідувач кафедри
професор Панчук В. Г.
(посада) (підпис) (дата) (ініціали та прізвище)

Рецензент

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

м. Івано-Франківськ — 2025 рік

Івано-Франківський національний технічний університет нафти і газу
(повне найменування закладу вищої освіти)

Інститут інститут інженерної механіки та робототехніки
Кафедра комп'ютеризованого машинобудування
Освітній рівень Бакалавр
Спеціальність 131 – Прикладна механіка
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри _____
_____
«____» _____20___року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Дрегало Олексій Юрійович
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка моделі марсоходу
керівник роботи Панчук В. Г., професор ,зав. кафедри КМВ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від "__" _____ 2025 року № ___/

2. Строки подання студентом роботи червень 2025р.

3. Вихідні дані до роботи: зразок матеріалу скляного волокна

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

1. Аналіз конструкції марсоходу Perseverance . 2. Модернізація моделі марсоходу. 3. Виготовлення моделі .

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Сумарним обсягом не менше 6 листів А1.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Панчук В. Г.,професор, зав. кафедри КМВ		
2	Панчук В. Г.,професор, зав. кафедри КМВ		
3	Панчук В. Г.,професор, зав. кафедри КМВ		

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Примітки
1	Аналіз марсоходу Perseverance	21.03.2025	
2	Модернізація моделі марсоходу	01.05.2025	
3	Виготовлення моделі	05.06.2025	
4	Захист магістерської роботи	25.06.2025	

Студент _____ Дрегало О.Ю.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Панчук В. Р.
(підпис) (прізвище та ініціали)

“___”_____2025 р.

Реферат

Бакалаврська робота на тему: «Проектування моделі марсохода». Дана робота складається зі 62 аркушів. До неї входять 61 рисунків, 2 таблиць, 16 додатків. Для розрахунку роботи було використано 20 бібліографічних найменувань.

Об'єкт дослідження: модель марсоходу як мобільна платформа для дослідницьких цілей.

Предмет дослідження: Конструктивно механічна реалізація моделі марсоходу, з урахуванням її функціональності та можливості подальшої модернізації.

Мета роботи: розробити, виготовити та вдосконалити конструкцію моделі марсоходу ,що буде функціональною та придатною для застосування в навчальних і демонстраційних цілях.

Основні завдання роботи:

- Аналіз конструкції марсоходу Perseverance, особливостей його шасі та підвіски.
- Оптимізація компонентів конструкції з урахуванням надійності, жорсткості та простоти складання.
- Створення тривимірної моделі корпусу та механічних вузлів у CAD-системі, підготовка STL-файлів для 3D-друку.
- Виготовлення моделі

Відповідно до поставленої задачі були виконані роботи за кожним розділом, згідно з планом:

1. Було проведено аналіз конструкції марсоходу з відкритих джерел . Виділено основні конструкційні вузли. Такі як підвіска типу Rocker-bogie ,будова коліс ,і рами .
2. Модернізація деталей марсоходу. Проаналізовано слабкі місця оригінальної моделі і виправши ці недоліки були для підвищення надійності і міцності .
3. Покращення електроніки і системи керування марсоходу, для збільшення модульності марсохода для подальших модифікація
4. Вибір матеріалів для 3д друку ,з рахуванням особливостей конструкції і технології 3д друку.

5. Підбір стандартних компонентів для збірки моделі.
6. Виготовлення моделі ,з пояснюванням головних моментів і вузлів з’єднання .

Студент Дрегало О.Ю.

Summary

Bachelor’s thesis on the topic: "Design of a Mars Rover Model" . This work consists of 62 pages. It includes 61 figures, 2 tables, and 16 appendices. A total of 20 bibliographic sources were used for the calculations and research.

Object of research: A Mars rover model as a mobile platform for research purposes.

Subject of research: The structural and mechanical implementation of the Mars rover model, taking into account its functionality and potential for further modernization.

Purpose of the work: To design, manufacture, and improve the structure of a Mars rover model that is functional and suitable for use in educational and demonstrational contexts.

Main objectives of the work:

- To analyse the design of the Perseverance rover, focusing on the features of its chassis and suspension system.
- To optimize the design components with respect to reliability, rigidity, and ease of assembly.
- To create a 3D model of the body and mechanical assemblies in a CAD system and prepare STL files for 3D printing.
- To manufacture the model.

In accordance with the stated objectives, the following tasks were completed as part of each section of the project:

1. The rover’s design was analyzed based on open sources. The main structural components were identified, such as the rocker-bogie suspension, wheel construction, and frame structure.
2. Modernization of the rover's components was carried out. Weak points of the original model were analyzed and corrected to improve reliability and strength.
3. The rover’s electronics and control system were improved to enhance modularity and allow for future modifications.
4. Materials for 3D printing were selected, taking into account the specific features of the design and the 3D printing technology.
5. Standard components were selected for assembling the model.
6. The model was manufactured, with explanations provided for the key assembly points and connection mechanisms.

Student: Drehalo O.Yu

Зміст

ВСТУП	9
1. АНАЛІЗ КОНСТРУКЦІЇ МАРСОХОДУ PERSEVERANCE	10
1.1 Загальні відомості про марсохід Perseverance	10
1.2 Ходова частина (підвіска Rocker-Bogie)	13
1.3 Конструкція коліс	16
1.4 Рама та несуча конструкція	19
2. МОДЕРНІЗАЦІЯ МОДЕЛІ МАРСОХОДУ	20
2.1 Загальний огляд базової моделі	20
2.2 Модернізація механічної конструкції	23
2.3 Оновлення електроніки керування	30
2.4 Оновлення програмного коду	35
3. ВИГОТОВЛЕННЯ МОДЕЛІ	42
3.1 Вибір матеріалів для друку та стандартних елементів	42
3.2 Особливості підготовки моделей до друку	44
3.3. Використання стандартних виробів і компонентів	47
3.4. Збірка механічної та електричної частини моделі	51
Висновки	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
Додатки	62

ВСТУП

Актуальність теми обумовлена зростаючою потребою в найрізноманітніших мобільних роботизованих систем. Вони застосовуються в найрізноманітніших сферах такі як : воєнна, агропромисловій ,цивільній, і в сферах надзвичайних ситуацій .Однак створення таких систем вимагають високих інженерних вмінь ,а також високих цін на комплектуючі. Тому розробка доступної ,модульної моделі марсоходу є актуальним завданням ,особливо для освітніх цілей.

Обєкт дослідженнія відкрита модель марсоходу Perseverance ,автора каналу YouTube "How to Mechatronics"

Мета: розробити, виготовити та вдосконалити конструкцію моделі марсоходу ,для навчальних і демонстраційних цілях.

Завдання:

- Аналіз конструкції марсоходу Perseverance
- Оптимізація компонентів конструкції з урахуванням надійності, жорсткості та простоти складання.
- Створення тривимірної моделі корпусу та механічних вузлів у CAD-системі
- Підготовка STL-файлів для 3D-друку.
- Модернізація електроніки керування
- Модернізація програмного забезпечення
- Виготовлення моделі

1.АНАЛІЗ КОНСТРУКЦІЇ МАРСОХОДУ PERSEVERANCE

1.1 Загальні відомості про марсохід Perseverance

Марсохід Perseverance місії Mars 2020 шукає ознаки давнього мікробного життя, щоб сприяти досягненню мети NASA — дослідженню минулої придатності Марса до існування життя. Ровер збирає кернові зразки марсіанських порід і реголіту (подрібнених порід і ґрунту) для можливого збору під час майбутньої місії, яка доставить їх на Землю для детального вивчення.[1]

Марсохід почали розробляти з грудня 2012 року ,запуск відбувся 30 липня 2020 року і приземлився на Марс 18 лютого 2021 року.

Рисунок 1.1 – Марсохід Perseverance на Марсі[4]

Ключове обладнання Perseverance має на борту сім наукових інструментів для проведення безпрецедентних досліджень та випробування нових технологій на Червоній планеті. Серед них:

- **Mastcam-Z** — передова камера з панорамним і стереоскопічним зображенням та можливістю зуму. Інструмент також визначатиме мінералогію марсіанської поверхні та допомагатиме в управлінні ровером. Головний дослідник — Джеймс Белл, Державний університет Арізони, Темпе.
- **SuperCam** — прилад, який може забезпечувати зображення, аналіз хімічного складу та мінералогії на відстані. Головний дослідник — Роджер Вієнс, Лос-Аламоська національна лабораторія , Нью-Мексико. Значний внесок також

зробили Centre National d'Etudes Spatiales, Institut de Recherche en Astrophysique et Planétologie (CNES/IRAP), Франція.

- **PIXL (Planetary Instrument for X-ray Lithochemistry)** — спектрометр рентгенівського флуоресцентного випромінювання та високоякісна камера для картографування дрібномасштабного елементного складу марсіанських матеріалів. PIXL забезпечить детальніше виявлення й аналіз хімічних елементів, ніж будь-коли раніше. Головна дослідниця — Абігайл Олвуд, Лабораторія реактивного руху NASA (JPL), Пасадена, Каліфорнія.

- **SHERLOC (Scanning Habitable Environments with Raman & Luminescence for Organics and Chemicals)** — спектрометр для тонкого зображення, який використовує ультрафіолетовий (УФ) лазер для картографування мінералогії та органічних сполук. SHERLOC стане першим УФ-раманівським спектрометром на поверхні Марса і доповнюватиме інші інструменти місії. Має також високоякісну кольорову камеру для мікроскопічного зображення марсіанської поверхні. Головний дослідник — Лютер Бігл, JPL.

- **MOXIE (Mars Oxygen In-Situ Resource Utilization Experiment)** — демонстрація технології, яка вироблятиме кисень із марсіанського вуглекислого газу. Якщо експеримент буде успішним, технологія MOXIE може бути використана майбутніми астронавтами на Марсі для створення пального для повернення на Землю. Головний дослідник — Майкл Хехт, Массачусетський технологічний інститут, Кембридж, Массачусетс.

- **MEDA (Mars Environmental Dynamics Analyzer)** — набір сенсорів для вимірювання температури, швидкості й напряму вітру, тиску, відносної вологості, а також розмірів і форми пилу. Головний дослідник — Хосе Родрігес-Манфреді, Centro de Astrobiología, Instituto Nacional de Técnica Aeroespacial, Іспанія.

- **RIMFAX (Radar Imager for Mars' Subsurface Experiment)** — радар для дослідження підповерхневих шарів з роздільною здатністю в сантиметри, що дозволяє вивчати геологічну структуру підповерхні. Головний дослідник — Свейн-Ерік Хамран, Norwegian Defense Research Establishment, Норвегія.

Корпус Perseverance та інші основні компоненти (такі як крейсерський модуль, етап спуску та аеродинамічний кожух/тепловий щит) створені на основі успішної конструкції ровера Curiosity і містять багато елементів, перевірених у попередніх місіях. Ровер Perseverance, розміром з автомобіль, має приблизно ті ж габарити, що й Curiosity: близько 3 метрів у довжину (без урахування маніпулятора), 2,7 метра в ширину та 2,2 метра у висоту. Однак Perseverance важить приблизно 1 025 кілограмів (2 260 фунтів), що на 126 кілограмів (278 фунтів) більше за Curiosity.

Perseverance також випробовуватиме нові технології для майбутніх роботизованих і пілотованих місій на Червону планету. Серед них — автопілот для уникнення перешкод під назвою *Terrain Relative Navigation*

(навігація відносно рельєфу)та набір сенсорів для збору даних під час посадки — *Mars Entry, Descent and Landing Instrumentation 2* (MEDLI2). Нова автономна система навігації дозволить роверу рухатися швидше по складному рельєфу.

Як і у випадку з Curiosity, основною енергосистемою Perseverance є *багатомісійний радіоізотопний термоелектричний генератор* (MMRTG), наданий Міністерством енергетики США. Він виробляє електроенергію за рахунок тепла, що утворюється при природному розпаді плутонію-238.[2]

Рисунок 1.2 Збирання марсоходу[2]

1.2Ходова частина (підвіска Rocker-Bogie)

В марсоходах використовується пасивний тип підвіски який називається Rocker-Bogie.Вона була розробленна NASA спеціально для марсоходів. Вперше її застосували в марсоході Sojourner(1997 рік).

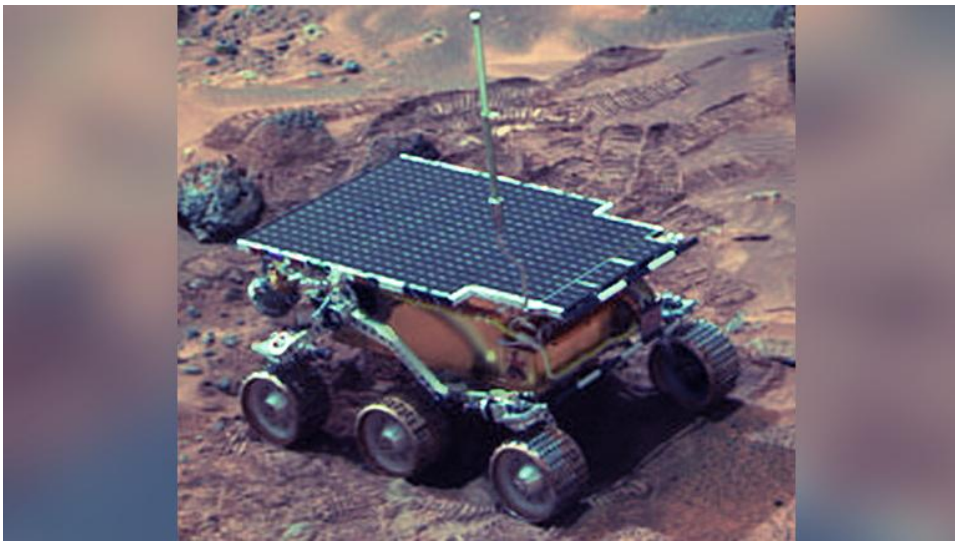


Рисунок 1.3 – марсохід Sojourner[5]

Ця підвіска складається з наступних компонентів:

- Rocker(гойдалка)(рис.1.4) це велика важільна частина яка з'єднує передні колесо і Bogie з центральною частиною .Ліва і права гойдали з'єднанні через диференціальний шарнір .

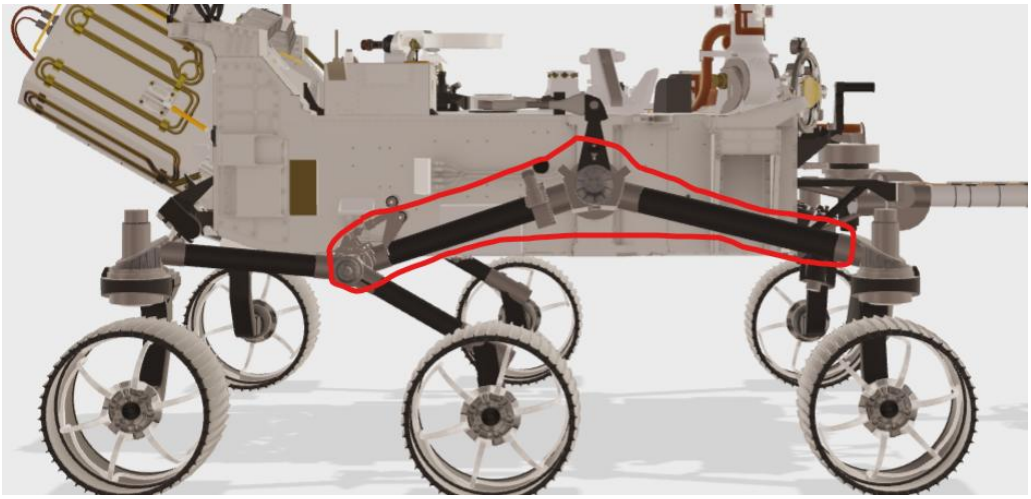


Рисунок 1.4 – Rocker ровера[10]

- Bogie(коромисло) – невелике з’єднання двох коліс яке приєднане до гойдалки, необхідне для адаптації і стабілізації відносно рельєфу .

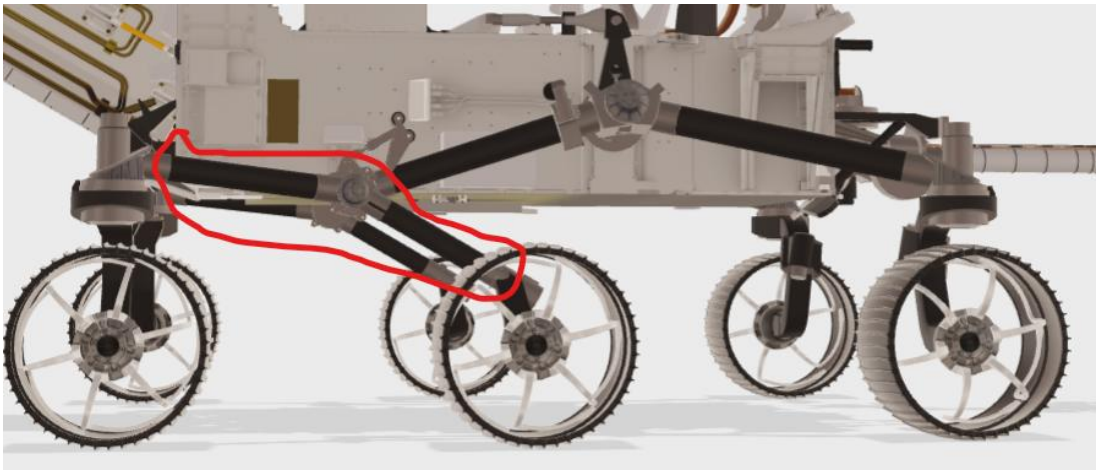


Рисунок 1.5 – Bogie ровера[10]

- 6 незалежних коліс які можуть обертатися незалежно одне від іншого ,також 4 країні колеса можуть обертатися що дозволяє обертатися на місці на 360 градусів

Принцип роботи криється в дії диференціала, яка дозволяє рівномірно розподіляти навантаження між колесами. Коли один бік підвіски піднімається (наприклад, на перешкоді), інший — опускається, забезпечуючи постійний контакт усіх шести коліс з поверхнею. Це зменшує тиск на ґрунт під окремими колесами — важливо, коли ровер рухається м’якою поверхнею, де надмірний тиск може призвести до «провалювання» колеса. Такий ефект також допомагає зберігати рівновагу: корпус ровера утримується під середнім кутом між двома гойдалками, зменшуючи нахил.[6]

NASA поєднує *rocker-bogie* з потужними мотор-редукторами в кожному колесі. Завдяки здатності системи утримувати всі колеса на землі, ровер має достатню тягу для подолання складного рельєфу. Навіть якщо одне чи два колеса зависли у повітрі або потрапили в пісок, інші компенсують це — і ровер не зупиняється.[6]

Ця проста, але надійна система без пружин і амортизаторів — ідеальна для умов Марса. Менше механічних частин означає менше потенційних поломок, менше зносу, менше шансів на забивання пилом. А це критично важливо, коли твій транспортний засіб знаходиться за мільйони кілометрів від технічного обслуговування.[6]

Єдина сфера, у якій конструкція *rocker-bogie* (гойдалка-балансир) поступається — це швидкість руху. Найшвидший марсохід NASA — *Perseverance* — має максимальну швидкість лише ~0.12 км/год, тоді як звичайна людська хода — близько 4 км/год. Така повільність пояснюється кількома причинами:

- **Затримка сигналу між Марсом і Землею** (від кількох до 20 хвилин) унеможливлює миттєве керування, а отже, швидкий рух безпечно реалізувати важко.
- **Низька швидкість зменшує ударні навантаження** — при високих швидкостях транспорт зазнає значних ударів від нерівностей. Щоб витримувати ці навантаження, потрібні складні демпфери (амортизатори), але *rocker-bogie* якраз від них відмовляється.
- **Простота конструкції** — повільний рух дозволяє зберегти конструктивну надійність і уникнути складних рішень.[6]

Рисунок.1.6 – Тестування підвіски в NASA JLP[7]

1.3 Конструкція коліс

Колеса в Perseverance пройшли значне поліпшення порівняно зі своїм попередником Curiosity після року роботи було виявлено значні пошкодження коліс що не були виявленні під час тестувань на Землі. Хоча колеса були спроектовані так щоб працювати навіть при значних пошкоджень але швидке зношування значно зменшувало очікуваний термін роботи коліс. Обробивши всі дані які були отримані з Curiosity ,поглибили розуміння рельєфу Марса ,а також його абразивної поверхні ,виникла потреба в значному поліпшенні конструкції коліс для майбутніх місій .

Під час посадки та переміщення колеса марсохода є першою структурою, що контактує з поверхнею Марса. Як наслідок, вони спроєктовані так, щоб діяти як пружина або демпфер навантажень з метою мінімізації зусиль, що передаються на основну конструкцію марсохода. Колеса також є єдиним елементом, що зменшує навантаження на приводи. Вони мають бути достатньо гнучкими, щоб забезпечити захист цих чутливих компонентів, але водночас досить міцними, щоб не зазнати катастрофічного руйнування під час посадки [8].

Марсохід MSL Curiosity та Mars 2020 Perseverance, як і їхні попередники, оснащені шістьма колесами та використовують підвіску типу "rocker-bogie". Ця підвіска дозволяє марсоходу долати перешкоди або заглиблення розміром до діаметра колеса, забезпечуючи кращу стійкість та менший кут нахилу.[8]

Кожне колесо складається з зовнішнього та внутрішнього ободів, кільця жорсткості, гнучких елементів (спиць) та ребер (ґраузерів), як показано на Рисунку 1.7. Гнучкі елементи забезпечують пружну підтримку колеса та приводів, а ґраузери — зчеплення у напрямку руху вперед, а також у поперечному напрямку. Ширина колеса також відіграє важливу роль у запобіганні його зануренню в ґрунт.[8]

Для марсоходів MSL Curiosity та Mars 2020 Perseverance кожне колесо виготовлено з блоку авіаційного алюмінію та оснащене титановими гнучкими елементами. Колеса Perseverance мають дещо більший діаметр порівняно з колесами Curiosity (20,7 дюйма проти 20,0 дюймів) та є трохи вужчими, при цьому товщина зовнішнього шару збільшена майже на міліметр. Також колеса мають нову конструкцію ґраузерів — злегка вигнуту замість шевронного малюнка, і їх кількість подвоєна (48 проти 24). Розширене тестування у Mars Yard Лабораторії реактивного руху NASA (JPL) показало, що нові протектори краще витримують тиск від гострих каменів і забезпечують таке ж або краще зчеплення на піску, а також зменшують ризик втомного руйнування, що спостерігався в конструкції Curiosity [8].

Рисунок 1.7 – Будова коліс Марсоходів[8].

Після успішного редизайну колеса виникла потреба покращити моделювання та точніше охарактеризувати поведінку колеса не лише під час руху по поверхні Марса, але особливо — під час посадки. У момент дотику до поверхні колеса піддаються значним деформаціям і напруженням. Внутрішнє дослідження JPL з моделювання посадкових навантажень показало: (1) існує невизначеність у врахуванні жорсткості колеса; (2) жорсткість колеса відіграє більшу роль у навантаженнях під час посадки, ніж очікувалося. У відповідь на це гнучкі елементи були оптимізовані на максимально можливу м'якість із збереженням структурної цілісності. [8].

Тому стало надзвичайно важливим точно визначити жорсткість і міцність нового колеса для підтвердження правильності розрахунків посадкових навантажень. [8].

Перероблене колесо марсохода Mars 2020 з оптимізованими гнучкими елементами було випробувано в JPL за допомогою візка механічного наземного допоміжного обладнання (MGSE), з гідравлічним циліндром, встановленим між ними,як показано на Рисунку 1.8. Установка була оточена опорною рамою для фотограмметрії, на яку були закріплені камери та освітлення. [8].

Статичні випробування колеса охоплювали загалом 32 навантажувальні випадки. Перші 30 призначались для визначення жорсткості, а останні два — для кваліфікації структури колеса. Випадки для визначення жорсткості були підібрані так, щоб охопити широкий спектр навантажень у різних точках колеса з метою повної характеристики жорсткості колісного вузла. [8].

Метою кваліфікаційного випробування було досягти 1,2-кратного граничного польотного навантаження (FLL) без шкоди для апаратури та 1,4-кратного FLL без катастрофічного руйнування випробуваного зразка. Пластина адаптера колеса змінювалася між навантажувальними випадками, щоб обертати колесо й відкривати різні точки кріплення, при цьому сенсор сили/моменту залишався нерухомим. Деталі щодо прикладення навантаження та розташування приладів показані на Рисунку 1.9. [8].

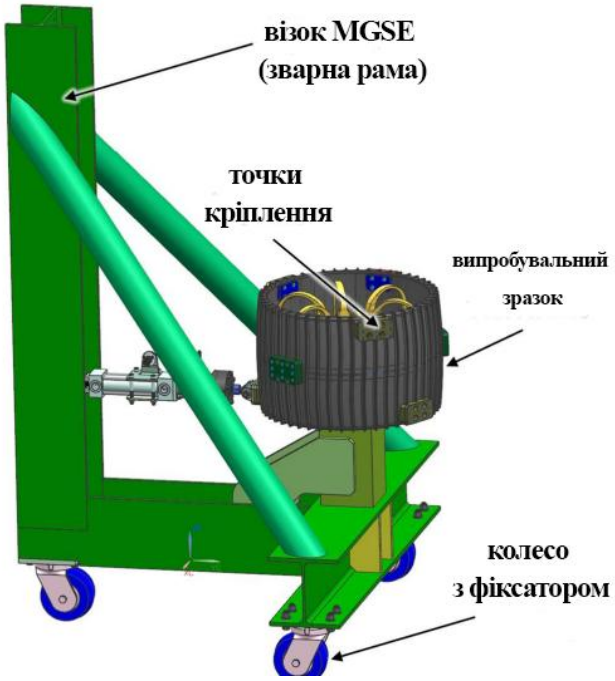


Рис.1.8 Візок MGSE з гідравлічним штоком і випробувальним зразком[8]

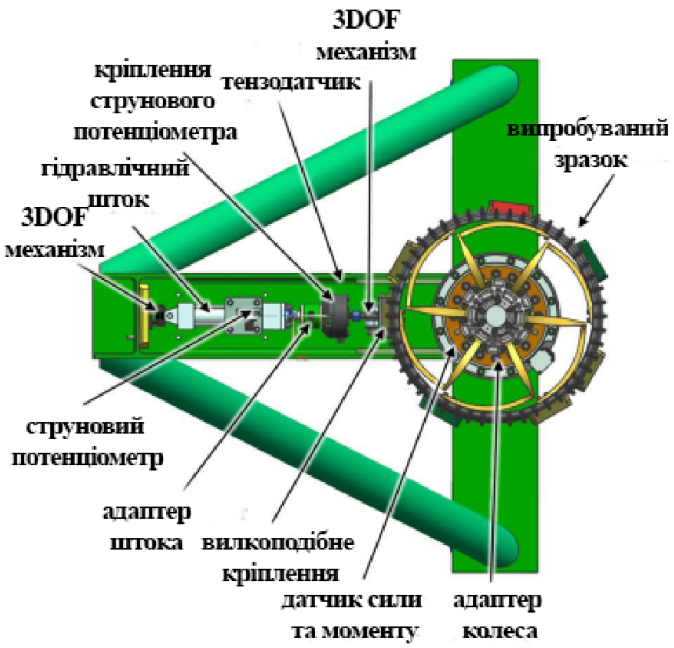


Рис.1.9 Застосування навантаження та навколишня апаратура (інструментування) [8]

1.4 Рама та несуча конструкція

Корпус марсохода Perseverance називається «тепловий електронний блок» (англ. *Warm Electronics Box*, скорочено – *WEB*)**.**Подібно до кузова автомобіля, корпус є міцною зовнішньою оболонкою, що захищає комп'ютер та електроніку марсохода — по суті, «мозок» і «серце» апарата.Корпус забезпечує захист і температурний контроль для всіх критично важливих систем марсохода.[9]

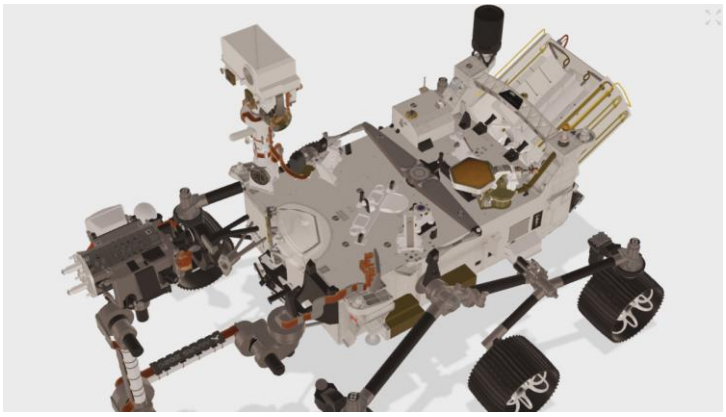


Рисунок 1.10 Корпус марсоходу[10]

Також корпус можна поділити на такі елементи :

- Основна рама - міцна і жорстка конструкція яка виготовлена з високоміцного алюмінієвого сплаву 7075 ,з'єднано це все титановими болти та вставки .Така конфігурація матеріалів дозволяє при невеликій масі витримувати сильні вібрації при посадці ,а також забезпечує довговічність при руху по нерівних поверхнях Марсу .
- Тепловий електронний блок – термоізоляційна частина рама яка крім того що захищає внутрішню електроніку від перепад температур ,а також від вібрацій і пилу ,виготовлена з багатошарової ізоляції зазвичай таке виготовляють з поліімідна фольга і каптону .
- Кронштейни підвіски жорстке кріплення для повороту Rocker-Bogie та передачі силових навантажень .Корпус виготовлений з титан Grade 5 ,а вісь обертання з загартованної нержавіючої сталі .
- Платформи для обладнання – яке має захист від вібрацій і деформацій .Виступає кріпленням для наукового обладнання ,камери ,антен і маніпулятора . Основа з алюмінієвий сплав (6061-T6) ,кріплення фіксатори ,вставки з титану.

2.МОДЕРНІЗАЦІЯ МОДЕЛІ МАРСОХОДУ

2.1 Загальний огляд базової моделі

За основу моделі марсоходу який розробляється для навчальних цілей ,було взято ,відкриту модель автора каналу і веб ресурсу “*How To Mechatronics”*.Він розробив 3д модель і програму для даної моделі .В подальших розділах буде розписана модернізація цієї моделі і процес її збірки .Але спочатку розглянемо цю базову моделі.

Рисунок 2.1 - марсохід “*How To Mechatronics”*[11]

Він використовує підвіску типу *rocker-bogie*, яка дозволяє плавно рухатися нерівною місцевістю та долати перешкоди, наприклад камені, розміром до подвійного діаметра колеса, зберігаючи при цьому контакт усіх шести коліс із поверхнею.Кожне колесо оснащене незалежним двигуном постійного струму (DC-мотором), що забезпечує рух марсохода вперед або назад. [11]

Рисунок 2.2 - підвіска rocker-bogie[11]

Чотири колеса по кутах марсохода оснащені індивідуальними сервоприводами для кермування. Щоб ефективно здійснювати повороти та уникати пробуксовки коліс під час руху по кривій, використовується кермова геометрія Акермана (*Ackermann steering geometry*). Завдяки цій геометрії можна обчислити швидкість і кут повороту кожного колеса залежно від радіуса повороту,Рисунок 2.3 . Це означає, що під час повороту внутрішні керовані колеса матимуть більший кут повороту порівняно з зовнішніми. Одночасно внутрішні колеса рухатимуться повільніше, ніж зовнішні. [11]

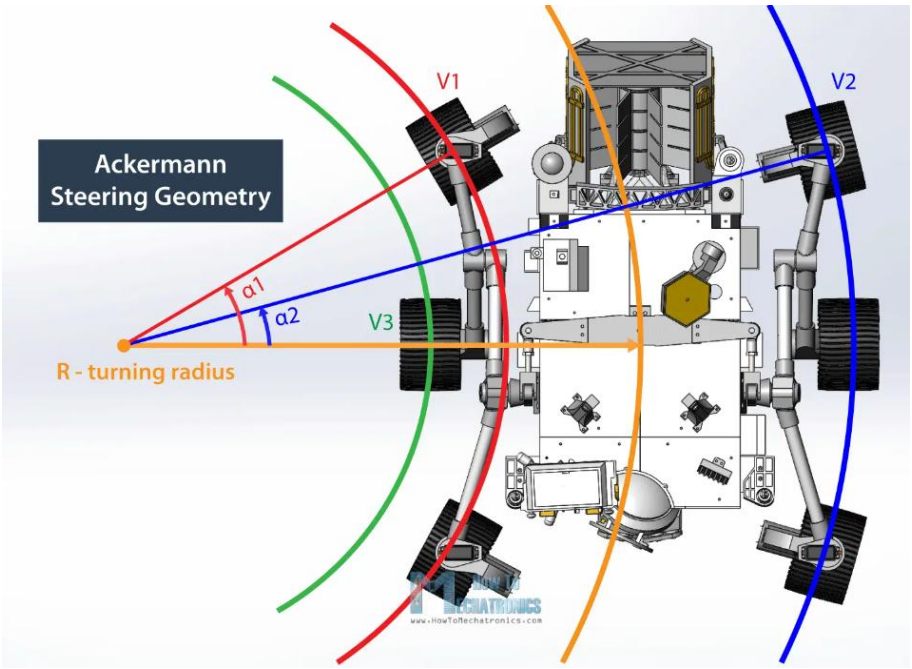


Рисунок 2.3 – Геометрія Акермана[11]

Ровер керується за допомогою радіо пульта,який передає сигнал на приймач на ровері який вже в свою чергу з'єдненний з Arduino MEGA яка є "Мозком" марсоходу. Arduino керує моторами і серво приводами . Моторами керуваня відбувається через драйвери DRV8871.Живлення схеми відбувається через LiPo акумулятор для захисту приймача і Arduino використовується DC-DC понижувач XL4016 .

Марсохід також оснащений FPV-камерою, розміщеною в блоці камер. Вона керується за допомогою крокового двигуна та сервопривода, а відео в реальному часі я отримую на смартфон. [11]

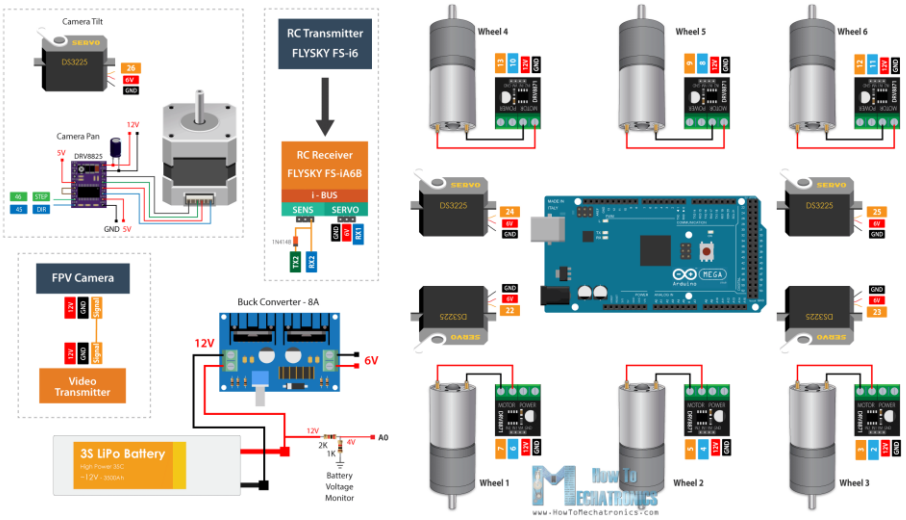


Рисунок 2.4 -схема підключення електроніки [11]

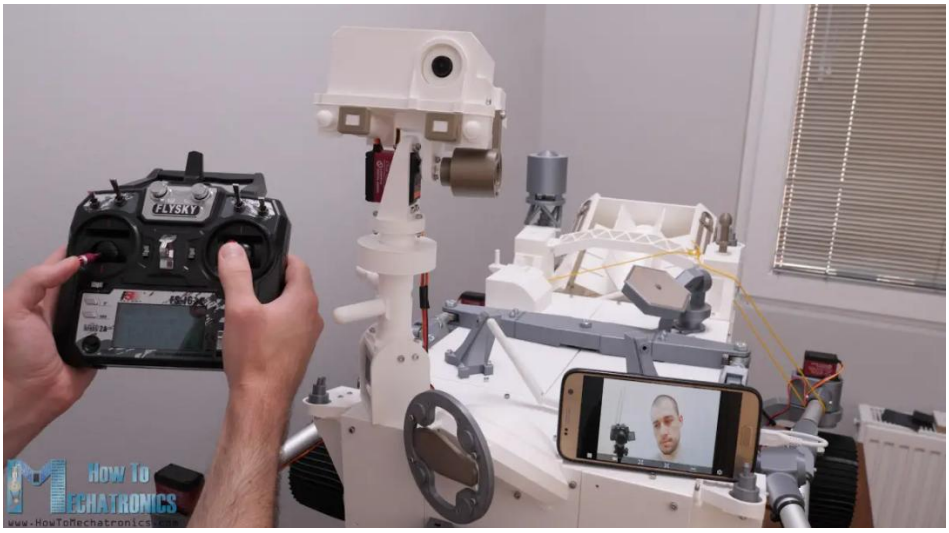


Рисунок 2.5 – Приклад роботи камери [11]

В цьому проекті автор зробив репліку яка включає не тільки основні елементи але й багато декоративних деталей .Моя задача стоїть в створенні основи яка вже зможе їздити і в подальшому яку можна модифікувати під різні задачі ,а також вивчати конструкцію в навчальних цілях .Тому модель яку я буду розробляти не буде мати таких елементів :

- Камери і відповідно конструкції під неї
- Декоративних елементів
- Обшивка рами буде відкритою

Завдяки такій конфігурації ровер чудово підходить для подальших модифікацій усіма, хто хоче глибше зануритися в цю тему.

2.2 Модернізація механічної конструкції

Під час збірки моделі було виявлено пару недоліків конструкції ,які в цьому розділі будуть описані і виправленні.

Для моделювання деталей було використано програмне забезпечення AUTODESK Inventor версії 2026.Це 3D САПР якій має широкій інструментарій для проектування і моделювання, а також надає безплатні ліцензії для студентів .

Почнемо з з'єднання сервоприводу ,тут проблема в двох деталях в кронштейні сервоприводу і його важеля .

Кронштейн в оригіналі має такий вигляд дивитися на Рисунок 2.6 і 2.7.

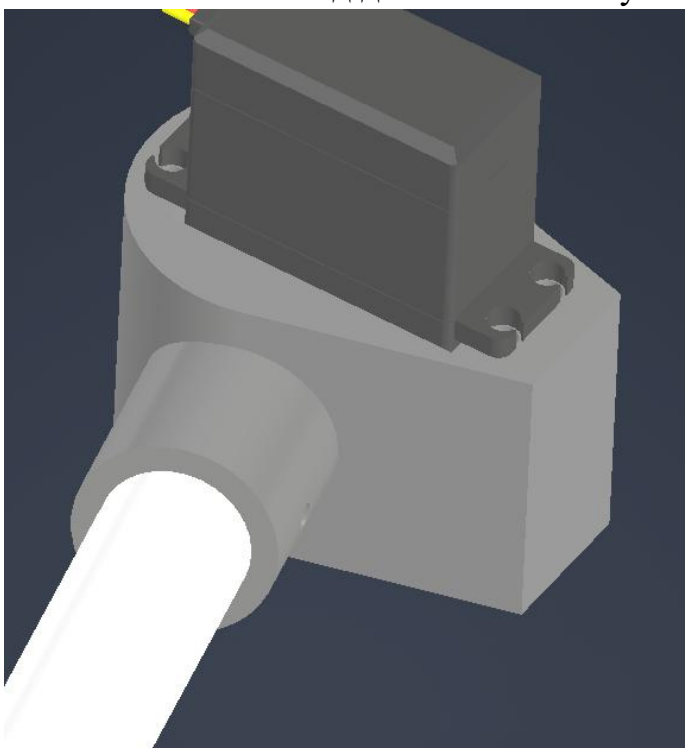


Рисунок 2.6 – кронштейн сервоприводу

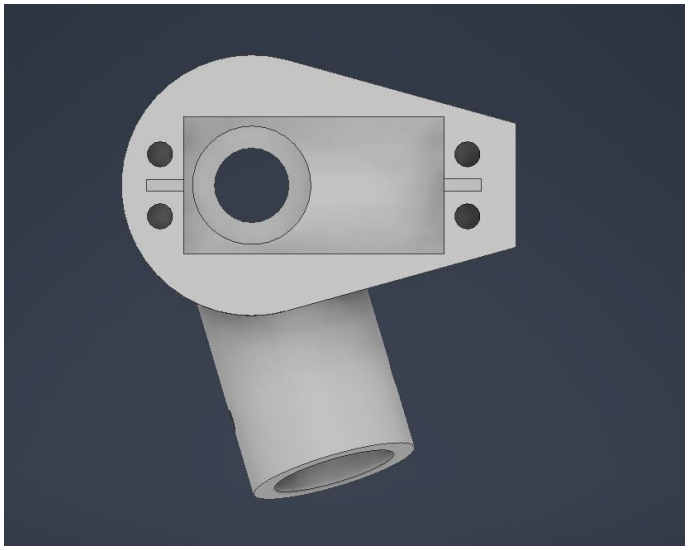


Рисунок 2.7 - кронштейн сервоприводу

Така конструкція кронштейну має недолік який не робить його надійним і не забезпечує виконання своєї функції .Основний недолік це те що саме серво не закріплено жорстко і немає нормального зчеплення з віссю обертання від цього виникають пропуски при обертанні .Тому перше рішення це створити спеціальне посадочне місце під серводвигун Рисунок 2.8.

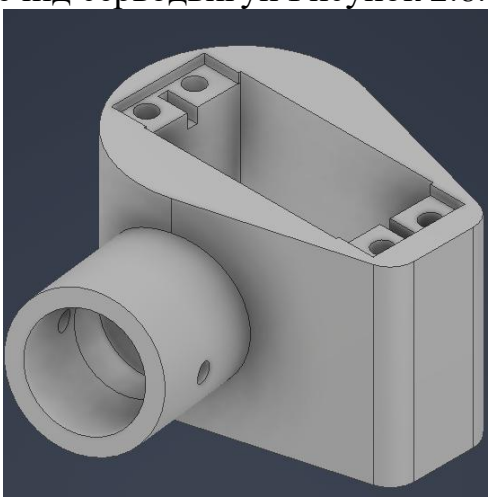


Рисунок 2.8 – посадкове місце

Таке поліпшення тримає сам сервопривід жорстко в кронштейні і не позволяє йому рухатися .

Друга проблема це великий отвір між підшипниками через що вісь обертання могла рухатися не строго перпендикулярно відносно колеса ,це видно на рисунку 2.7.Рішення зменшити отвір до діаметру осі обертання ,плюс невеликий допуск на зазор. Результат на рисунку 2.8

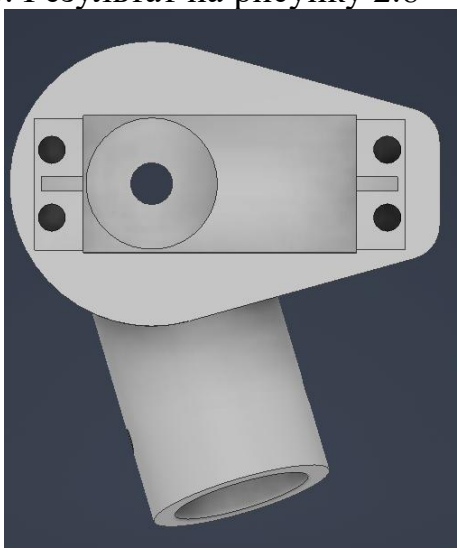


Рисунок 2.8 – зменшений отвір під ось обертання

Третя проблема вже стосується особливості друку на FDM принтерах. Кронштейн кріпиться до підвіски через алюмінієву трубу .Отвір під цю трубу друкується горизонтально ,тому він спотворюється сильно і труба не сідає в нього. Рішення значно збільшити допуск посатки на зазор ,0.5мм, також в кінець отвору зробили конічним з ще більшим зазором .Ці зазори не вплинуть на лишній небакенний рух труби ,так як сама труба закріплюється болтом і гайкою самоблок. Результат на рисунку 2.9.Креслення цієї деталі є в додатку .

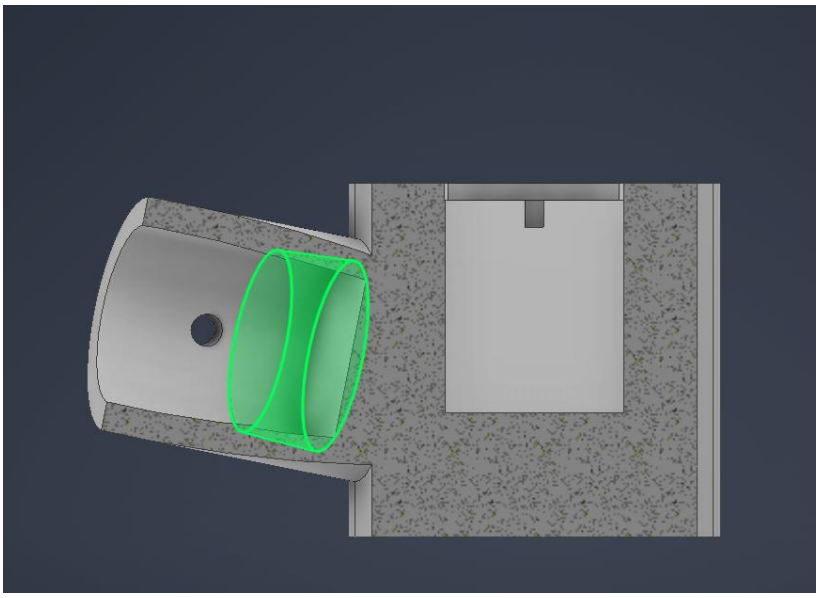


Рисунок 2.9 – Отвір під кріплення труби

Наступне розберемо які проблеми були в важиля сервоприводу .Основна його проблема це в допусках і посадках які не відповідають те що треба щоб закріпити його на сервоприводі ,а також встановити вісь обертання .Рішення просте збільшити розміри під отвори до кроштейну серводвигуна і також збільшити розмір шестикутника під вісь обертання яка являє собою болт М8 з плоскою головкою .Результат на Рисунку 2.10 , а також повне робоче креслення в Додатку.

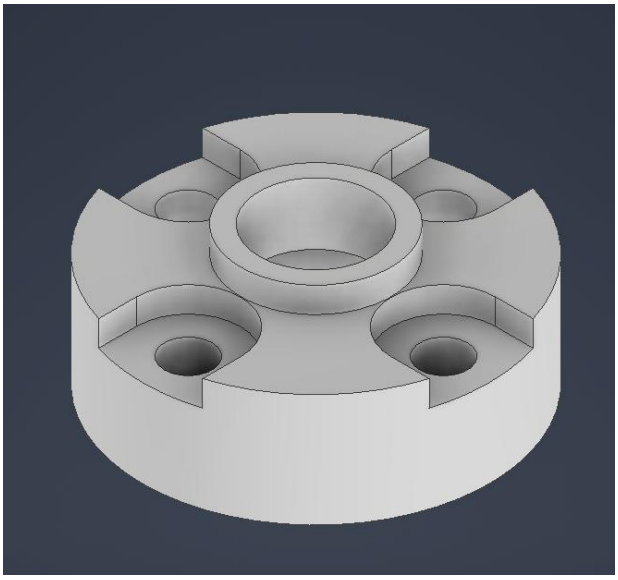


Рис.2.10 важіль сервоприводу

На рисунку 2.11 зображенно порівняння між версіями ,зліва оригінальна, з права

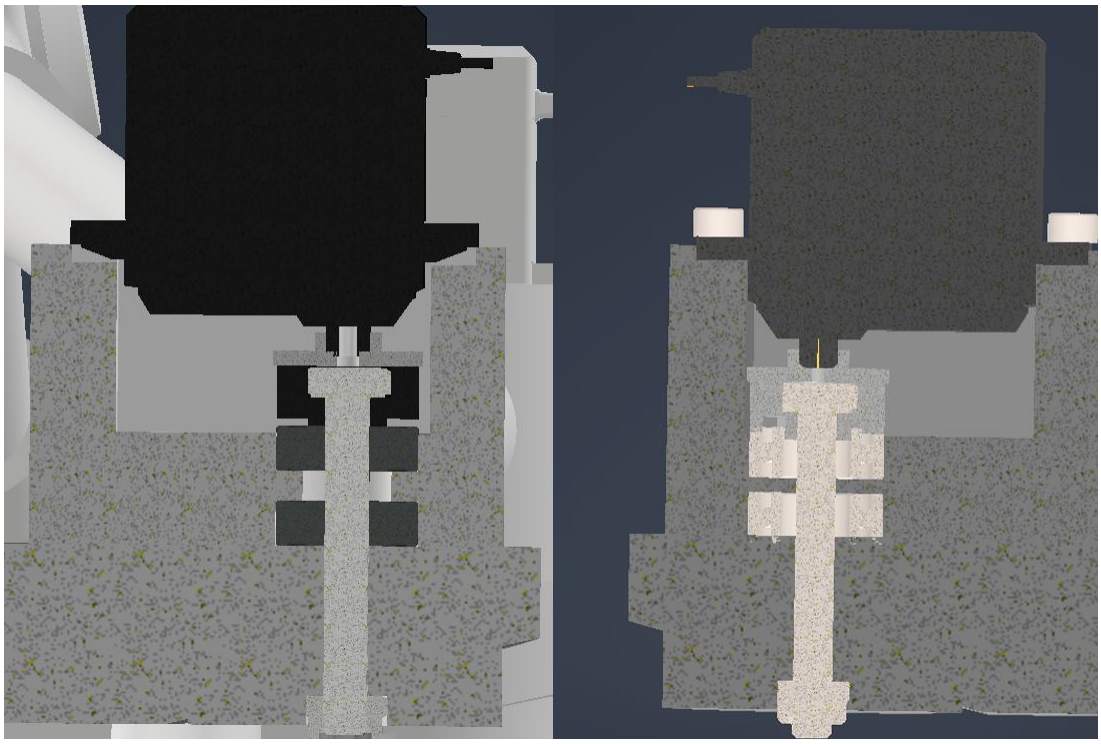


Рисунок .2.11 – Порівняння двох варіантів

Наступна деталь на розгляді муфта яка з’єднує обід колеса з двигуном постійного струму .На рисунку 2.12 зображена оригінальна деталь .

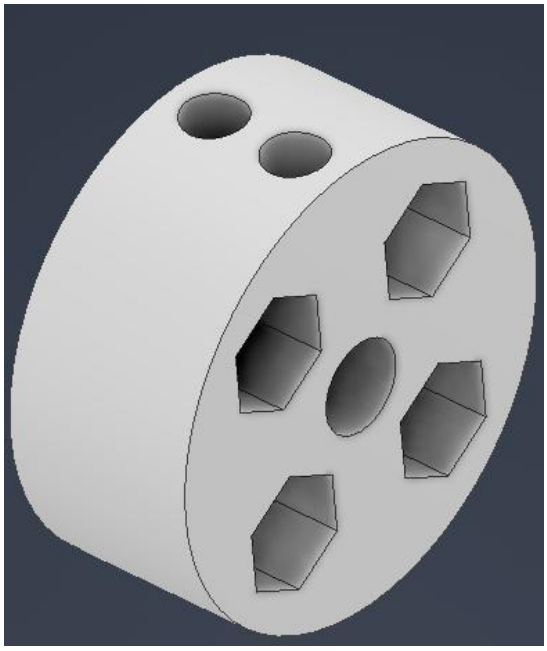


Рисунок 2.12 муфта

Тут Дві проблеми :

1. Не відповідний допуск деталі що робить не можливим без додаткової обробки насадити на двигун а також вставити гайки в відповідні пази

2. По задумці тут мали б використовуватися установчий гвинт М4 ,але не було вказано якої він довжини і стандарту ,плюс ще можна додати те що в таких гвинтах кінець не плоский а конічний і не забезпечує міцного і надійного зчеплення з валом ,через що може крутитися мотор без муфти .

Тому рішення наступні :

1. Збільшити допуск посадок отворів
2. Замість установчих гвинтів використати звичайні болти М3 стандарту DIN 912 і сховати головки в муфті

Результат на рисунку 2.13 ,а також робоче креслення є в додатку

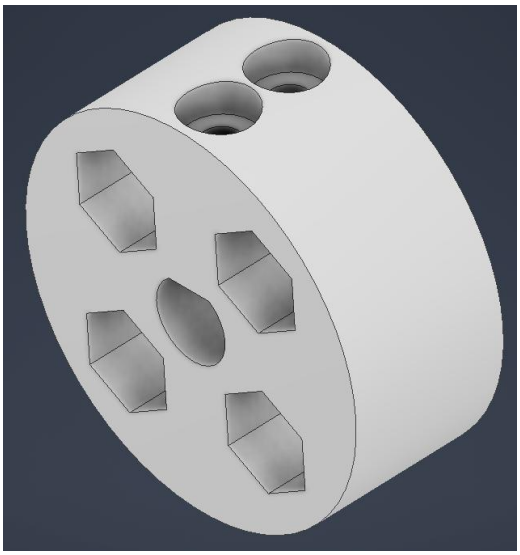


Рисунок 2.13 - модифікована муфта

Наступна деталь на це кронштейнів для алюмінієвих верстатних профілів .

Під час збірки було виявлено що стандарті кути для верстатних профілів ,приклад зображений на рисунку 2.14 ,не забезпечує потрібну жорсткість рами ,плюс ще одну причину те що вони не дуже широкодоступний товар через що ціна відповідна.

Тому мною було спроєктовано кронштейн в вигляді пластини з стінками для надійної фіксації профілів ,і яку можна легко і швидко надрукувати на 3д принтері. Є 2 модифікації цієї пластини під різне з’єднання яка використовується ровері:

Рис.2.14 - Алюмінієвий кут для верстатного профілю 2020

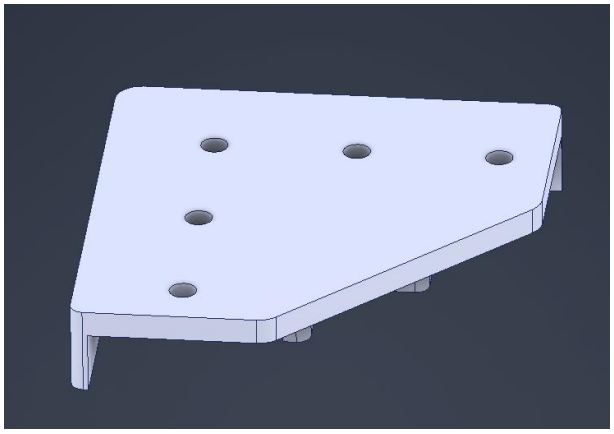


Рисунок 2.15 – кутовий кронштейн

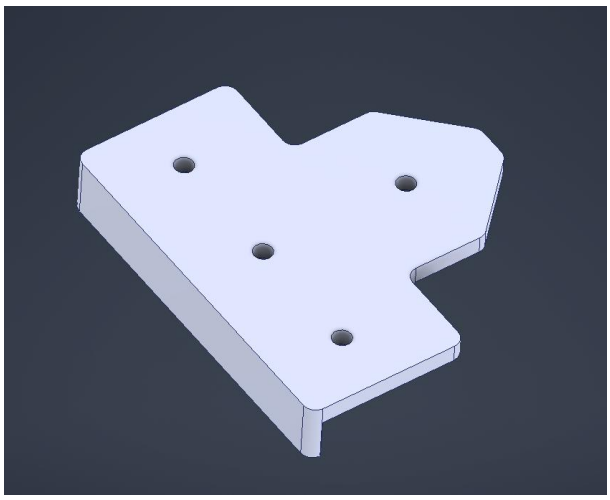


Рисунок 2.16 – т-подібний кронштейн

Як має вигляд з’єднання з рамою зображено на рисунку 2.17 і 2.18.

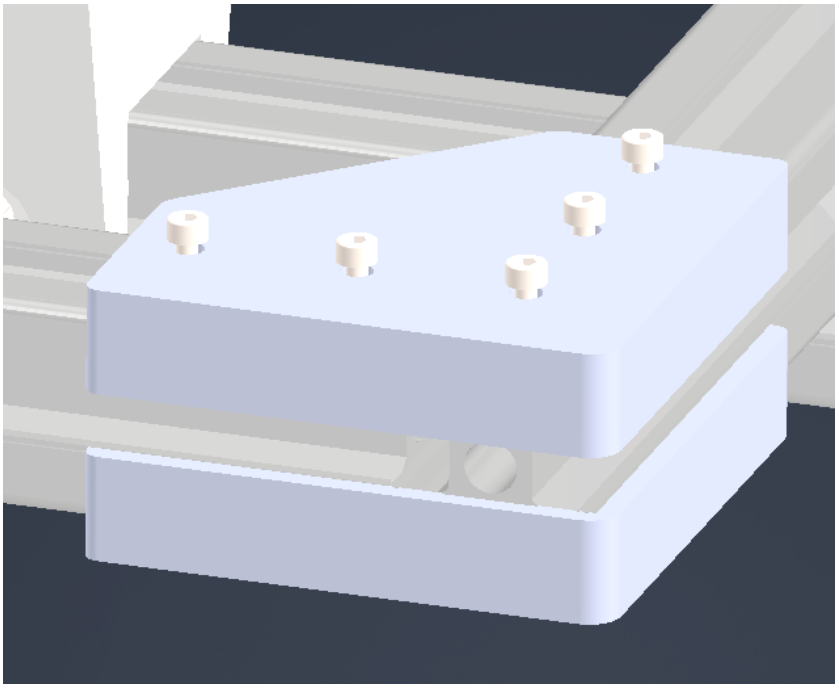


Рисунок 2.17 - з’єднання з рамою кутового кронштейну

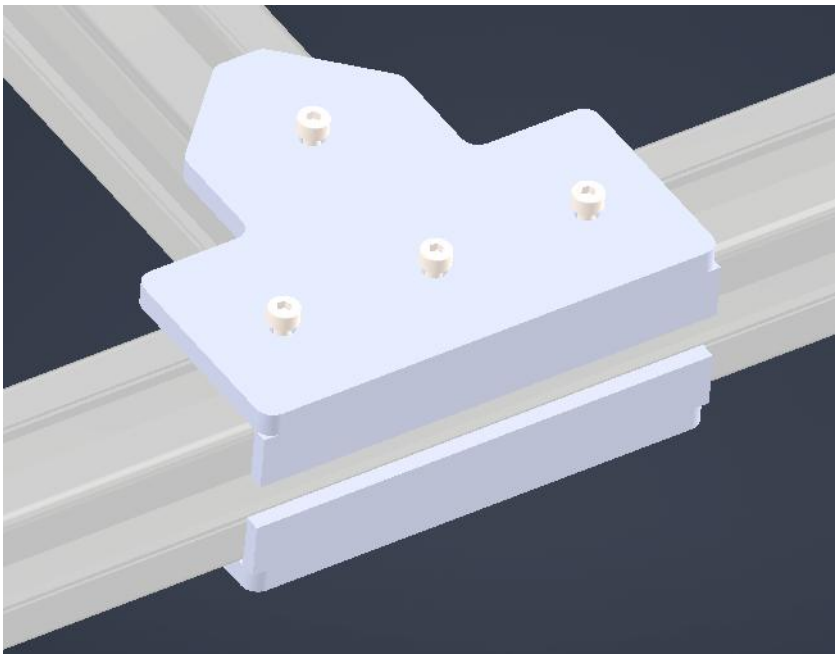


Рисунок 2.18 – з’єднання з рамою т-подібного кронштейну

2.4 Оновлення електроніки керування

В цьому підрозділі буде описано оновлення електроніки керування . В оригінальній моделі використовується в ролі “мозку” ровера Arduino Mega ,яка на даний час вже морально застаріла. Як в плані обчислювальних потужностей ,так і плані можливостей .Тому прийнято рішення замість ардуїно використовувати мікроконтролер сімейства Esp32 ,а саме плату розробника ESP32-WROOM Devkit V1(рисунок 2.19).Вона переважає Arduino Mega майже всьому ,також її головна особливість це те що в ній є вбудований модуль Wifi і Bluetooth.Що значно розширює можливості застосування плати в різних задач .Але і має свій негативний вплив в вигляді високого споживання енергії .Тому якщо треба високо енергоефективність треба розглядати інші варіанти. Але в випадку ровера ,збільшення споживання електроенергії не грає суттєвої ролі .В таблиці 2.1 наведено порівняння по основних характеристиках мікроконтролерах Arduino і Esp32.

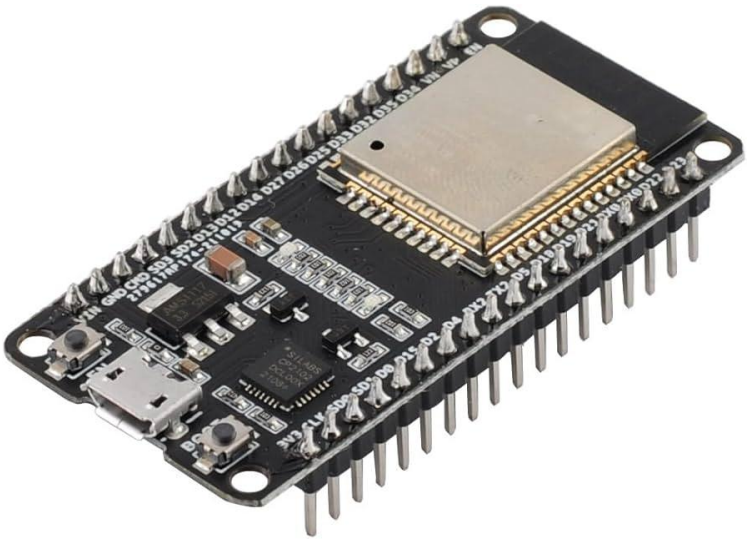


Рисунок 2.19 – плата розробника ESP32[12]

Таблиця 2.1[13]

	ESP32	Arduino Mega2560
Флеш-пам'ять	4 МБ	256 КБ
SRAM	520 КБ	8 КБ
EEPROM	-	4 КБ
Тактова частота	240 МГц	16 МГц
Робоча напруга	3.3 В DC	7-12 В DC
Споживання струму	80 мА – 90 мА	150 мкА
Цифрові входи/виходи	36	54
Аналогові входи	18	16
ШІМ	16	14
UART	3	4
I2C	2	1
SPI	4	1
CAN	1	-
Wi-fi	Стандарт IEEE 802.11 b/g/n	-
Bluetooth	Bluetooth: v4.2 (BR/EDR, BLE)	-

Проект задовольняє потреби нашого проекту але є можливість розвинути і використати Wifi зв'язок замість радіо зв’язку що позволить керувати ровером з любого ПК ,телефона який має доступ.

Друга ідея поліпшення це зменшення використання пінів мікрокотролера шляхом переведення керування на через плату посередника .Для цього було використано 16-канальний 12-бітний ШІМ/Серво драйвер на мікросхемі PCA9685 від компанії Adafruit.Це плата керується через інтерфейс протокол I2C ,відповідно дозволяє 2 проводами керувати 16 сервоприводами або драйверами моторів .В I2C є недолік в його швидкості обміну данних але в нашому випадку дані це не складні протоколи передачі данних і ровер сам по собі не рухається швидко і йому висока швидкодія інших протоколів як UART ,SPI або CAN не потрібна .

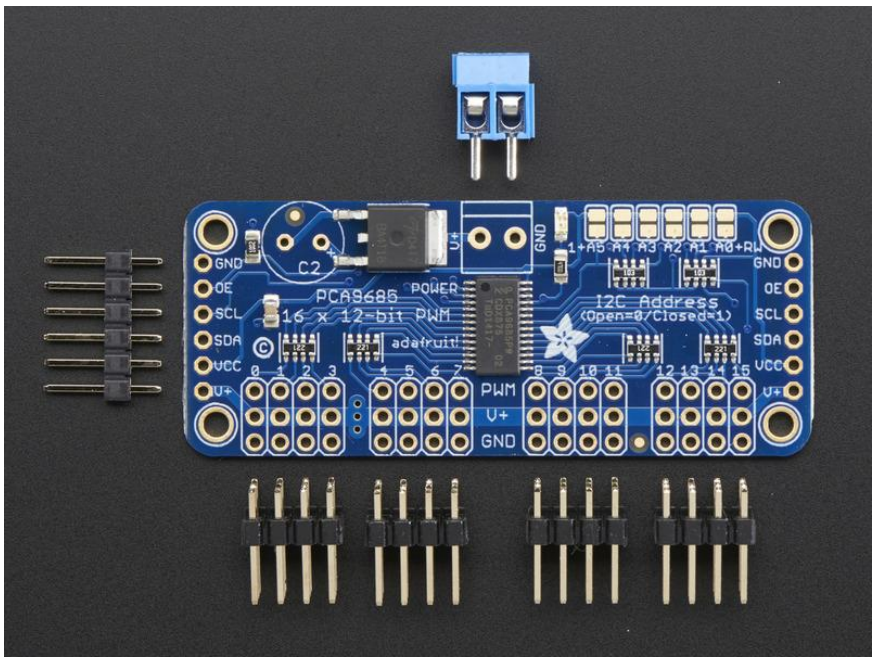


Рисунок 2.20 - драйвер на мікросхемі PCA9685 від Adafruit[14]

В проекті це плата керує сервоприводами і також встановлює швидкість моторів . таким чином було зекономлено 10 пінів мікроконтролера. Схема підключення зображено на Рисунку 2.21.

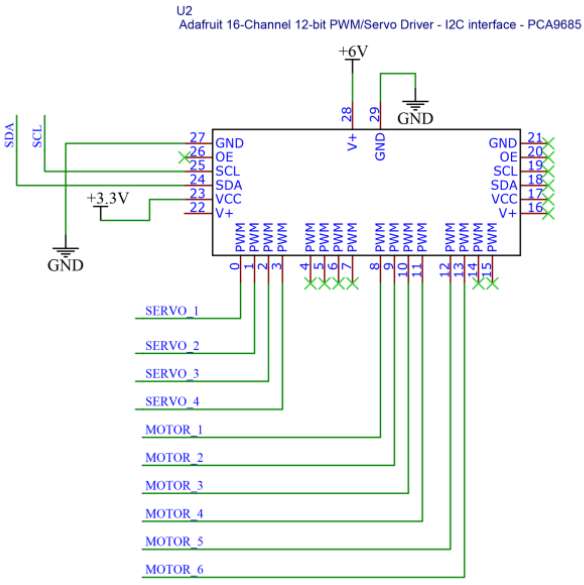


Рисунок 2.21 – схема підключення драйвера

Наступне що було змінено це драйвер для керування моторами. В проекті автора “How to mechatronics” використовується драйвер на мікросхемі DRV8871.Це плата повністю задовлільняє потреби моделі для керування моторами ,але має деякі недоліки.Перший недолік що ця плата є досить дорога і може коштувати більше ніж мікрокотролер esp32.Другий її недолік є як і плюсом так і мінусом ,це проста логіка керування .Керується двома контактами які встановлюють напрям обертання і швидкість.З таким керуванням кладно реалізувати складну логіку роботи коліс.

Тому це було замінено на драйвер двоканальний на мікросхемі TB6612FNG. Він позволяє керувати одразу 2 двигунами ,задовольняє силові потреби в живленні моторів, має малу ціну .Цей драйвер має вдосконалену логіку керування ,має різні режими : гальмування ,зворотне обертання ,пряме обертання , Стоп ,очікування . в таблиці 2.2 наведена таблиця істинності для встановленні різних режимів .

Таблиця 2.2

IN1	IN2	PWM	STBY	OUT1	OUT2	Опис режиму
1	0	-	1	0	0	Гальмування
0	1	1	1	0	1	Зворотне обертання
0	1	1	1	0	0	Гальмування
1	0	1	1	1	0	Пряме обертання
1	0	0	1	0	0	Гальмування
0	0	1	1	0	0	Стоп
-	-	-	0	0	0	Очікування

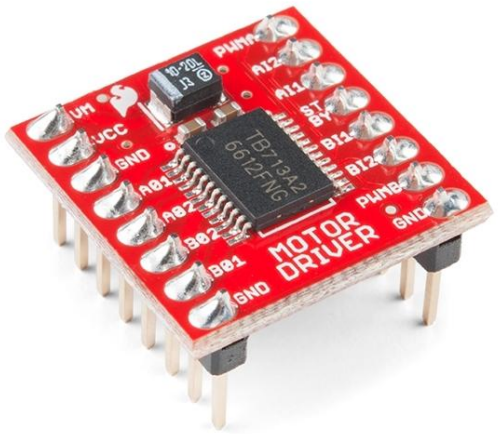


Рисунок 2.22 – Двоканальний драйвер моторів на TB6612FNG[15]

На рисунку 2.23 зображено схема підключення одно з драйверів ,в загальному в блоці керування використовується 3 таких драйвери.

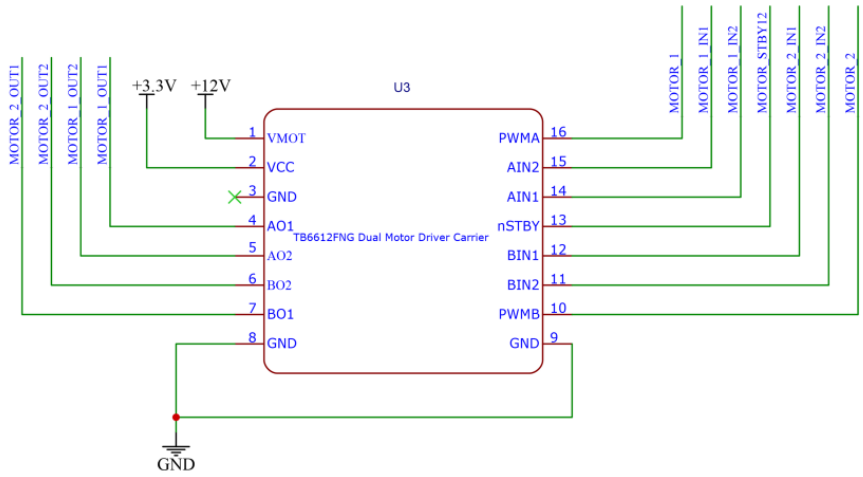


Рисунок 2.23 – Схема підключення драйвера TB6612FNG

Як висновок після цих поліпшень блок керування став дешевший в виготовленні, функціонал і обчислювальні потужності зросли завдяки заміні мікроконтролера .Також “делегуванні” керування сервомоторами і ШІМ сигналом драйверу PCA9685 вдалося зекономити контакти мікроконтролера для використанні в подальших модифікацій .Заміна драйверів для двигунів постійного струму розширила можливості логіки без застосування складних алгоритмів також знизала вартість всього блоку. Загальна електрична принципова схема розміщена в додатку.

2.4 Оновлення програмного коду

В попередньому розділі було описано модернізацію електроніки блоку керуванні і відповідно таке оновлення вимагає оновлення програмного коду для нього. Програмне забезпечення будемо писати на Arduino Framework на мові програмуванні C++.

Раніше популярним рішенням для написання коду було використання Arduino IDE ,а зараз воно є застарілим засобом розробки який не позволяє зручно і швидко писати програми для мікроконтролерів. Тому для цього буде використовувати сучасніший підхід ,використовувати середовище VS code з розширенням PlatformIO.

PlatformIO — це кросплатформний, кросархітектурний, багатофреймворковий професійний інструмент для інженерів вбудованих систем і розробників програмного забезпечення, які пишуть застосунки для вбудованих продуктів.[16].Цей інструмент має широкий функціонал :

- Система управлінням проектами .Кожний проєкт має конфігураційний файл *platformio.ini* в якому можна налаштувати :
    - Плату або MCU(board)
    - Компілятор/платформу(platform)
    - Фреймворк(framework)
    - Прапорці компіляції (build_flags)
    - Параметри логування ,інтерфейс прошивання і так далі
- Стандартна структура проєкту як в більшості інших сфер програмування
- Пітримує найрізноманітніші платформи(Arduino,ESP32 ,STM32, AVR,Rasbery Pi і інші) і фреймоворки(Arduino Framework , ESP-IDF ,STM32Cube ,FreeRTOS , CMSIS і інші)
- Вбудоване Юніт-тестування для хосту(native) так і на самому залізі
- Керування бібліотеками ,легко можна встановити обновити бібліотеку для потреб любого проєкту
- Пітримує різні види моніторингу : серійний монітор ,фільтри ,логування в файл
- Інтеграція з різними середовищами розробки : VS code , CLion ,Atom ,Eclipse
- Різні види інтерфейсів налагодження (GDB ,ST-Link ,J-Link,OpenOCD)

Як видно по списку можливостей зверху ,можна повністю впенитися що цей інтрумет на декілька порядків кращий за звичайний Arduino IDE ,навіть для

написання коду під саме Arduino.

Дальше розглянемо саму структуру коду .Проект складається з таких головних елементів :

- Файл конфігурації проєкту platformio.ini в якому встановлюється тип плати ,фреймворк ,встановлюємо швидкість моніторингу і підключаємо необхідні бібліотеки . файл має наступний вигляд:

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
monitor_speed = 115200
lib_deps =
        blynkkk/Blynk
              adafruit/Adafruit PWM Servo Driver
        Library@^3.0.2
```

- Файл config.h включає в собі константні змінні налаштувань контактів для сервомоторів ,фрагмент коду:

```
//конфігурування каналів керування
#define SERVO_1 0
#define SERVO_3 1
#define SERVO_4 2
#define SERVO_6 3

#define MOTOR_STBY12 13

#define MOTOR_1 8
#define MOTOR_1_IN1 14
#define MOTOR_1_IN2 12

#define MOTOR_2 9
#define MOTOR_2_IN1 27
#define MOTOR_2_IN2 26
```

- І головний файл коду програми main.cpp в якому розміщена вся логіка прошивки ,його ми розберемо детально .

Головний файл можна розділити на такі під блоки : підключення бібліотек , оголошення змінних ,блок функцій ,блок налаштування і блок вічного циклу .

В проекті використовуються дві основні бібліотеки : “Adafruit_PWMServoDriver.h” і “BlynkSimpleEsp32.h”

“Adafruit_PWMServoDriver.h” -бібліотека для керування 16 канальним драйвером ШІМ/Серво на мікросхемі PCA9685 ,через протокол I2C.В проекті використовуються наступні функції з цією бібліотеки :

- Для робити з драйвером необхідно створити об’єкт його класу :

```
// Драйвер серво PCA9685
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
```

- В функції PCA9685_init використовуються функції базового ініціалізації драйвера .Функції :begin для ініціалізації і перевірки чи правильно підключений драйвер до мікроконтролера ,setPWMFreq функція встановлення частоти виходи з драйвера для сервоприводів і моторів ,код функції наведений нижче :

```
// Ініціалізація PCA9685
void PCA9685_init(int servoInitAngle)
{
  // Ініціалізація PCA9685
  if (!pwm.begin())
  {
    Serial.println("Помилка ініціалізації PCA9685!");
    while (true); // Зупинка, якщо ініціалізація не вдалася
  }
  else
  {
    Serial.println("PCA9685 успішно ініціалізовано.");
  }

  pwm.setPWMFreq(50); // Частота для сервоприводів (50 Гц)

  setServoPWMViaPCA9685(SERVO_1, servoInitAngle);
  setServoPWMViaPCA9685(SERVO_3, servoInitAngle);
  setServoPWMViaPCA9685(SERVO_4, servoInitAngle);
  setServoPWMViaPCA9685(SERVO_6, servoInitAngle);
}
```

- Функція setServoPWMViaPCA9685 встановлює довжину імпульсу ШІМ сигналу для сервоприводу ,в ній використовується функція з бібліотеки setPWM ,в ній є 3 параметри : перший це номер каналу в якій треба встановити ШІМ ,наступний рівень імпульсу на початку і останній рівень імпульсу в кінці :

```
// Встановлення ШІМ для серво
void setServoPWMViaPCA9685(uint8_t servoChannel, int microseconds)
{
   uint16_t pulseLength = map(microseconds, 0, 180, SERVO_MIN, SERVO_MAX);
   pwm.setPWM(servoChannel, 0, pulseLength);
}
```

Наступна бібліотека на огляд - “BlynkSimpleEsp32.h”.

Blynk – це повністю інтегрована платформа Інтернету речей, яка не залежить від обладнання. Вона має мобільні додатки, хмарні сервіси і дозволяє керуванням пристроями, аналітикою даних і машинним навчанням.[17]

Блінк є це повним набором програмного забезпечення, необхідного для створення прототипу, розгортання та віддаленого керування підключеними електронними пристроями в будь-якому масштабі: від персональних проектів IoT до мільйонів комерційних підключених продуктів.[17]

Платформа забезпечує практично усе, що вам потрібно для створення підключеного обладнання та керування ним. Це підготовка (provisioning) пристроїв, візуалізація даних датчиків, дистанційне керування за допомогою мобільних і веб-додатків, оновлення мікропрограми по повітрю (OTA), безпечна хмара, аналіз даних, керування користувачами та доступом, сповіщення, автоматизація та багато іншого більше.[17]

Хоча цей сервіс більше призначений для інтернет речей ,але в додатку є віджети для які позволить нам керувати ровером. На рисунку 2.24 зображенний інтерфейс керуванням ровера :

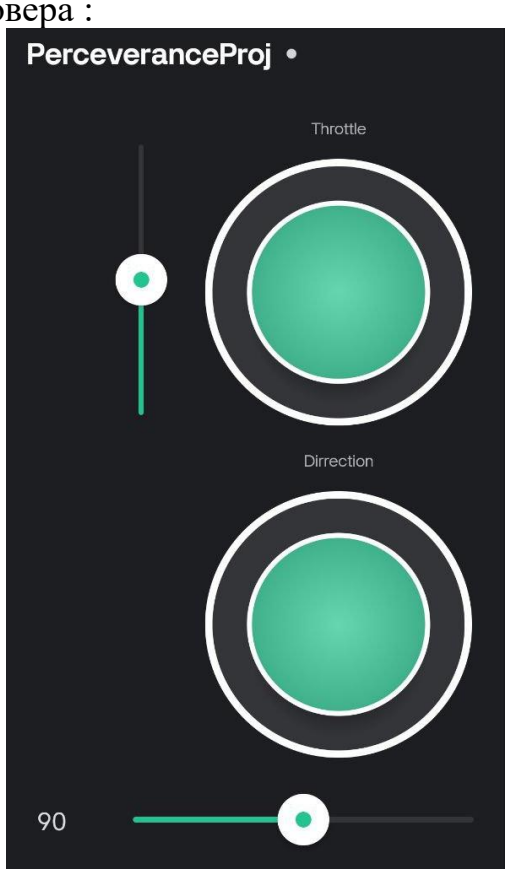


Рис.2.24 – Інтерфейс керування в Blynk App

Цей інтерфейс є простим але дозволяє замінити радіопульт і позволить керувати моделлю з будь-якого смартфону або комп'ютера в якого є доступ до акаунту Blynk ,Тепер розглянемо як реалізувати це керування в коді :

- Для початку треба підключитися до Wifi і до серверів Blynk ,для цього треба такий перелік константних змінних : BLYNK_TEMPLATE_ID , BLYNK_TEMPLATE_NAME , BLYNK_AUTH_TOKEN , auth , ssid ,pass.Тепер ми можемо підключитися за допомогою функції ініціалізації :

```
Blynk.begin(auth, ssid, pass);
```

- Наступне треба визначити функції зворотного зв'язку з сервера ,які працюють по принципу переривання і встановлюють нове значення змінної швидкості і кута :

```
//Функція виклику Blynk V1
BLYNK_WRITE(V1)
{
  XPlot = param.asInt(); // Отримати значення кута з повзунка (0-180)
}

//Функція виклику Blynk V2
BLYNK_WRITE(V2)
{
  YPlot_b = param.asInt(); // Отримати значення кута з повзунка (-90-90)
}
```

- Для того щоб зв'язок тримався стабільно треба в вічному циклі програми викликати функцію Blynk.run()

Тут показані всі етапи необхідні щоб зв'язати ровер з сервісами Blynk ,як видно це все просто і не потребує великих знань щоб це реалізувати.

Наступним розглянемо як реалізовані алгоритми обчислення для кінематики марсоходу .Алгоритм керування реалізований за кермова геометрія Акермана.

В ровера є 4 колеса які повертаються тому треба робити 4 розрахунки два для внутрішніх і два для зовнішніх коліс .Згідно геометрії зробимо формули 2.1…2.4 ,приставка до змінної In означає внутрішні колеса ,Out - зовнішні ,F - переднє ,B - заднє :

$$\text{angle\_In\_F} = \tan^{-1}\left(\frac{D_3}{r+D_1}\right) \times \frac{180}{\pi} \qquad (2.1)$$

$$\text{angle\_In\_B} = \tan^{-1}\left(\frac{D_2}{r+D_1}\right) \times \frac{180}{\pi} \qquad (2.2)$$

$$\text{angle\_Out\_F} = \tan^{-1}\left(\frac{D_3}{r-D_1}\right) \times \frac{180}{\pi} \quad (2.3)$$

$$\text{angle\_}0ut\text{\_B} = \tan^{-1}\left(\frac{D_2}{r-D_1}\right) \times \frac{180}{\pi} \quad (2.4)$$

Сталі величини ланок підвіски : D_1=271 мм D_2=278 мм D_3=301 мм D_4=304 мм .Змінна r - це загальний радіус на який має повернутися колеса ,є вхідної змінної величиною який отримується від пульта.В коді ці формули мають наступний вигляд :

```
// Обчислити кут для кожного сервоприводу на основі заданого радіуса повороту
void calculateServoAngle(int radius)
{
  thetaInnerFront = round((atan((LINK_D3 / (radius + LINK_D1)))) * 180 / PI);
  thetaInnerBack = round((atan((LINK_D2 / (radius + LINK_D1)))) * 180 / PI);
  thetaOuterFront = round((atan((LINK_D3 / (radius - LINK_D1)))) * 180 / PI);
  thetaOuterBack = round((atan((LINK_D2 / (radius - LINK_D1)))) * 180 / PI);
}
```

Дальше треба обчислити швидкість коліс ,є два сценарії якщо кут повороту в межах 90 то це означає що модель знаходиться в прямому положенні отже швидкість всіх пар коліс однакова ,в інакшому випадку ,є 3 швидкості. Це обчислюється за допомогою формул 2.5…2.7 :

$$s_1 = s \quad (2.5)$$

$$s_2 = s \times \frac{\sqrt{D_3^2 + (r-D_2)^2}}{r+D_4} \quad (2.6)$$

$$s_3 = s \times \frac{r-D_4}{r+D_4} \quad (2.7)$$

Величина s це змінна величина швидкості яка отримується від пульта .В коді обчислення швидкості має наступний вигляд :

```
// Обчислення швидкості двигунів
void calculateMotorsSpeed(int radius, int speed)
{
  // Якщо немає кермового впливу, швидкість усіх коліс однакова — прямолінійний рух
  if (YPlot > 89 && YPlot < 91)
  {
    speed1 = speed2 = speed3 = speed;
  }
  // Під час повороту швидкість коліс залежить від значення радіуса повороту
  else
  {
    // Зовнішні колеса, найдальші від центру повороту, мають найбільшу швидкість
    // Через геометрію ровера всі три зовнішні колеса повинні
    // обертатися майже з однаковою швидкістю.
    // Різниця становить лише близько 1%, тому вважаємо їх однаковими.
    speed1 = speed;
    // Внутрішні передні та задні колеса ближче до точки повороту і мають нижчу
    // швидкість порівняно із зовнішніми колесами
    speed2 = speed * sqrt(pow(LINK_D3, 2) + pow((radius - LINK_D1), 2)) / (radius + LINK_D4);
    // Внутрішнє середнє колесо найближче до точки повороту і має найнижчу швидкість
    speed3 = speed * (radius - LINK_D4) / (radius + LINK_D4);
  }

  // Значення швидкості від 0 до 100% конвертується у PWM від 0 до 4095
  speed1PWM = map(round(speed1), 0, 100, 0, 4095);
  speed2PWM = map(round(speed2), 0, 100, 0, 4095);
  speed3PWM = map(round(speed3), 0, 100, 0, 4095);
}
```

3.ВИГОТОВЛЕННЯ МОДЕЛІ

3.1Вибір матеріалів для друку та стандартних елементів

Виготовлення моделі почнемо з вибору матеріалів з яких вона буде зроблена. Вибір матеріалів спираються на особливості конструкції і її як вона буде використовується а також на обмеження 3Д друку. Також беручи до уваги що модель призначена для навчальних цілей,треба додати критеріями були доступність матеріалів, легкість і доступність технічної обробки та достатня механічна міцність для імітації реальної конструкції.

В моделі використовуються в більшості використовується для 3д друку філамент PLA .Його головна перевага це доступність і легкість друку ,також він забезпечує необхідну міцність і жорсткість для моделі,але у в усіх вузлах моделі. Наприклад в підвісці є одне місце сили зависокі і PLA не витримує ,тому треба використати більш міцніший матеріал ABS ,на рисунку 3.1 з чорного пластику виготовлений шарнір хитуна .

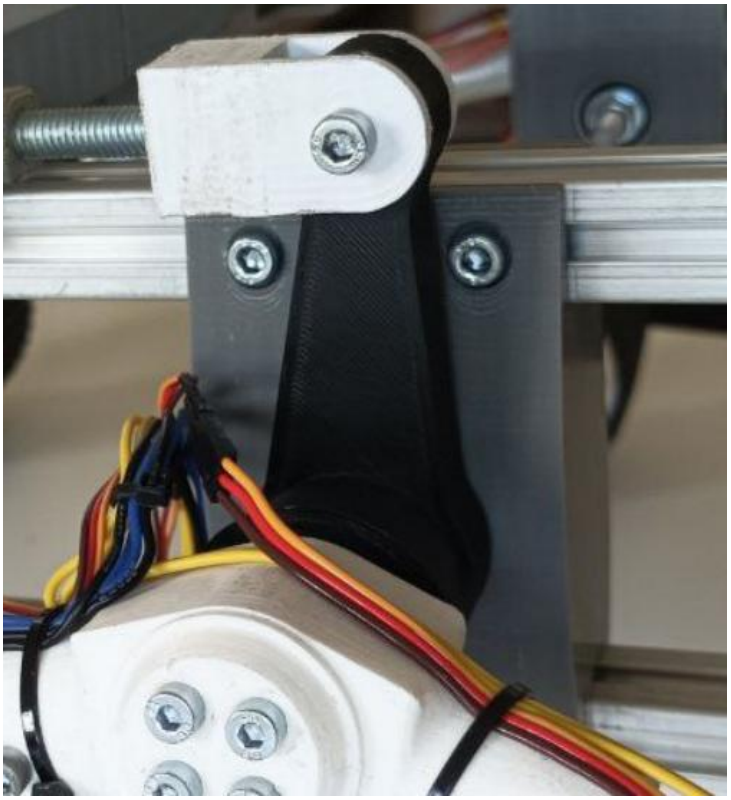


Рис.3.1 – шарнір хитуна з ABS

Також в ровера є потенціал до заміни деяких деталей з PLA на PETG , він не був зроблений переважно з нього по причині коли друкувалися основні деталі цей філамент не був сильно поширений а також він вимагає вищих температур що не кожний принтер зміг би з ним впаритися .Іншою проблемою є те що самі колеси також зроблені з PLA але він досить жорсткий для такої деталі ,тому краще було б надрукувати його з гнучкого пластику такий як TPU ,але через досить великі вимоги до самого 3д принтера і величину самої деталі ,друкування такої деталі вимагає принтер промислового рівня, тому треба було приймати компромісне рішення.

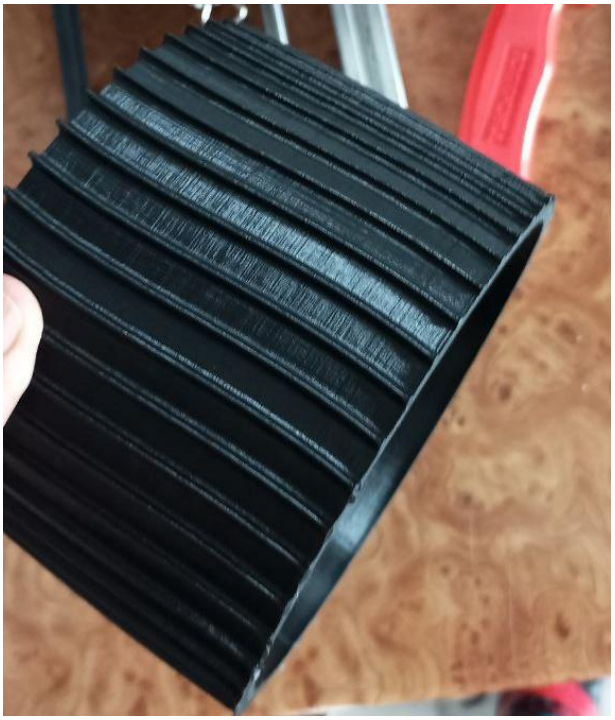


Рисунок 3.2 – Колесо виготовлене на 3д принтері PLA пластиком

Крім пластикових деталей в моделі використовуються стандарті деталі такі як алюмінієві труби і верстатні алюмінієві профіля 20х20 .Причина їх вибору дуже проста, вони є міцними легко доступними і легко обробляються .

3.2 Особливості підготовки моделей до друку

Виготовлення деталей на 3д принтері є один з складних і найдовших етапів всієї розробки і виготовлення моделі ,це зумовлено тим що 3д принтер хоч і дозволяє виготовлювати дешево складної форми деталі ,але в нього також є недоліки такі як :

- Мала швидкість виготовлення деталей, деякі деталі друкувалися більше 10 годин
- Деталі не виходять точними і треба робити додаткову механічну обробку
- Довга підготовка ,кожну деталь вимагає правильного коригування вихідного файлу для друку ,а також треба підготувати сам принтер для роботи ,перевірити подачу пластику рівень стола і так далі
- Великий процент браку ,навіть якщо все було перевіренно на етапі підготовки ,брак може сягати до 30%

Але не дивлячись на всі ці недоліки виготовлення моделі є набагато легшим і дешевшим якщо б це робилися традиційними методами виготовлення .Також ця індустрія не стоїть на місці і постійно розвивається що більшість недоліків які я перерахував вище значно зменшилися за останні роки.

Основна більшість деталей було виготовленна 3д принтері VERNERFAB i3 v 2.1.

Рисунок 3.3 – принтер VERNERFAB i3 v 2.1.[18]

Принтер VERNERFAB i3 2.1 — це модифікована версія популярної конструкції Prusa i3, орієнтована для вивчення 3D-друку.Кінематична схема типу Cartesian(Декартива) ,екструдер типу Bowden ,робоча зона 220x220.

Також використовувся принтер Flying Bear Ghost 5 – це можна сказати вже наступне покоління доступних принтерів .Цей принтер перевершує попредній в усіх параметрах має більшу область друку (255x210 мм) ,має міцну металеву раму яка зменшує вібрації ,сучасне програмне забезпечення і багато іншого .

Рисунок 3.4 – принтер Flying Bear Ghost 5 [19]

Для підготовки моделей необхідний слайсер ,для виготовлення моделі марсоходу було використано Ultimaker Cura. Це безкоштовне програмне забезпечення для підготовки і нарізанні моделі на g-code для 3д принтера. Розгялнемо підготовку деталі до друку на прикладі обіду колеса .Парметри на які треба звернути для підвищення міцності : товщина стінок(wall thickness) ,товщина низу(Bottom thickness) і верху(Top thickness) і процент заповнення (infill destiny).Також треба звернути на температуру для філаменту ,цей параметре відрізняється від виробника пластику тому треба слідувати рекомендаціям ,а також дещо змінювати відносно того як друкує сам принтер .Також деякі моделі для успішного друку треба друкувати пітримки(support).Отже щоб надрукувати деталь до колеса були зміненні такі параметри :

- Товщина стінок 1.6 мм
- Товщина низу 0.8 мм
- Товщина верху 0.8 мм
- Процент заповнення 50%
- Температура хотенда 225 ℃
- Температура стола 65 ℃
- Підтримки включенні

Такі параметри мають баланс між міцність і швидкості друку .На рисунку 3.5 зображенне вікно в Cura з готовою моделлю яку можна розглянути пошарова як буде друкуватися деталь ,стільки займе часу виготовлення і витрати по пластику в грамах і м. З часом в Cura є проблема ,він не відображає дійсну затрату принтера для деталі .Наприклад обід колеса мав би друкуватися 7 годин 46 хвилин ,а в реальності цей час треба множити на 1.25 і 1.35 щоб приблизно оцінити реальний час друку .

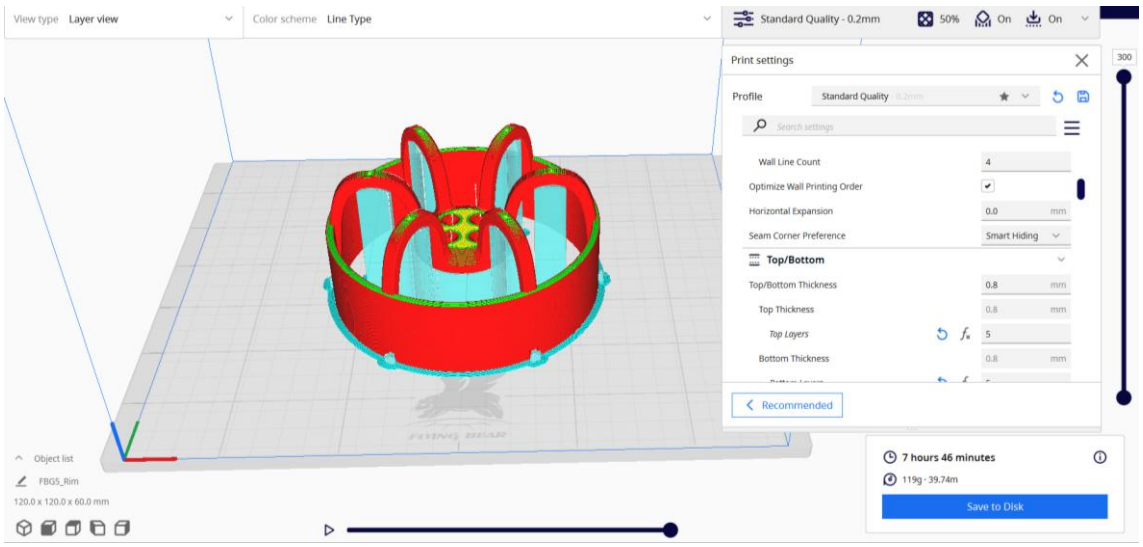


Рисунок 3.5 – “нарізана” модель для друку обіду колеса

Решта деталей друкувалися з подібними параметрами .Вийняток становить деталі виготовлення з ABS.Друк цим пластиком для відкритого типу принтерів ,є справжнім випробуванням ,цей пластик має більшу міцність ніж PLA але під час друку він є набагато слабшим ,і любий потік лишнього повітря може заставити модель розійтися по шарам пластику ,тому крім підняття температури хотенда до 250 ℃ і стола 90℃ ,також треба зменшити швидкість друку до 50мм/с,і один з самих головних параметрів це відключення охолодження моделі(Enable print cooling) ,також через те що модель яка друкувалася з ABS це шарнір хитуна ,повинна на себе сприйняти великі навантаження процент заповнення збільшенний на 75%.

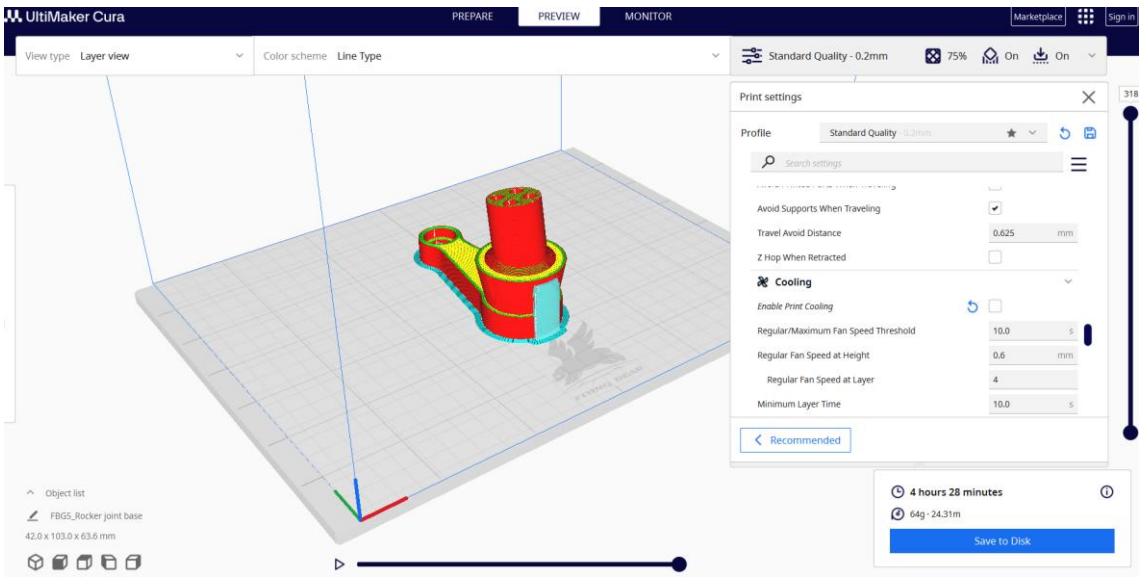


Рисунок 3.5 підготовлена модель до друку шарніра хитуна

3.3Використання стандартних виробів і компонентів

В моделі використовується великий список стандартних виробів ,дальше коротко розглянемо основні кріпильні компоненти. В основному всі болти стандарту DIN 912 ,через їх міцність ,легкість монтування і доступність .Крім гвинтів також застосовуються різьбові стержні ,вони використовуються щоб з’єднати диференціальний шарнір з шасі і рамою .Гайки представленні 3 видами : звичайні по типу DIN 934 ,само блокуючі Din985 і Т-подібні для алюмінієвих профілів 20х20 .Шайби використовуються в з’єднані частин шасі ,щоб розподілити навантаження ,а також зменшити тертя на пластикові деталі.

Рисунок 3.6 – Гайки самоблок М8 DIN 985

Рисунок.3.7 – Болти М5х38

Для рухливих ланок використанні підшипники кулькові однорядні стандарту DIN 6xxRS ,різних діаметрів.Також використовуються шарнір кульковий наконечник стандарту ISO 12240-4.

Рисунок 3.8 – Підшипник DIN 625RS

Рисунок 3.9 – Шарнір кульковий наконечник ISO 12240-4 [20]

В проекті в ролі основи рами рами використовуються верстатні алюмінієві профілі 20х20 ,забезпечують необхідну жорсткість і міцність рами .

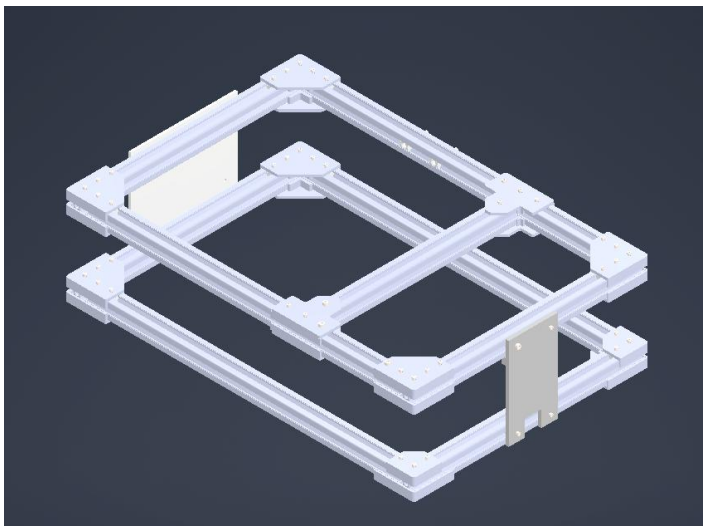


Рисунок 3.10 – конструкція рами марсохода

З’єднання ланок в Rocker-Bogie реалізоване через алюмінієві труби діаметром 20.

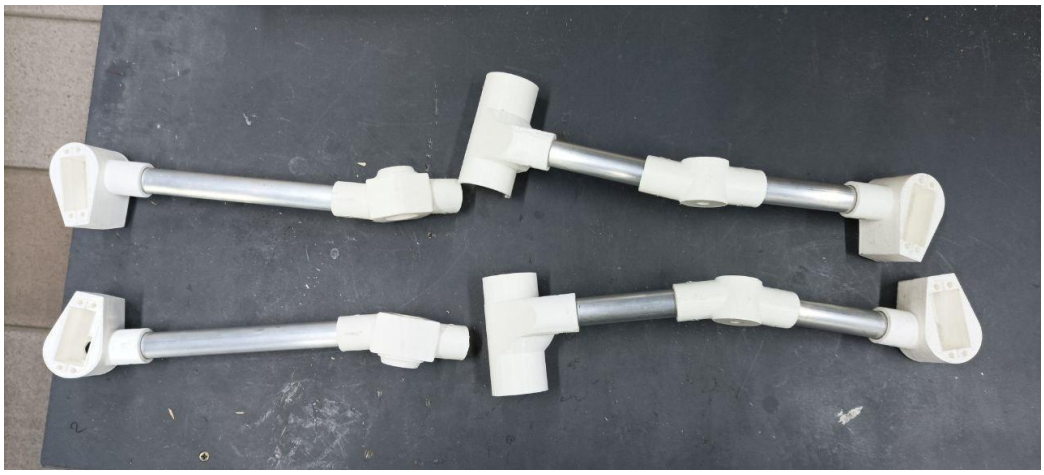


Рисунок 3.11 – частина шасі марсоходу

В ровері використовується редукторні електродвигуни XD-37GB555 з можливістю регулювання їх швидкості. Вони мають напругу живлення 12В,максимальний струм 3А ,номінальний струм 0.6-1.5А ,потужність 15Вт.Діаметр двигуна 37мм ,діаметр вала 6 мм, циліндричний тип редуктора, мають високий крутний момент ,50 обертів на хвилину .

Рисунок 3.11 – редукторний електродвигун XD-37GB555[11]

В марсоході використовуються серводвигуни DS3230 Pro ,для здійснення поворотів. Серводвигун має наступні характеристики : робоча напруга 6-8.4 вольта ,крутний момент до 30кг*см, металеві шестерні з сталі ,розмір 40.5x20x40.5,керування через ШІМ сигнал з частотою 50Гц.

Рисунок 3.12 – серводвигун DS3230 встановлений на марсоході

3.4.Збірка механічної та електричної частини моделі

В цьому розділі я опишу основні етапи збірки моделі марсоходу.

Почнемо з збирання трьох опорної ланки шасі особливість збірки цієї ланки полягає в тому щоб всі отвори через які з’єднуються ланки були на одній лінії ,це можна легко зробити притиснувши трубу і провести олівцем ,а потім зробити отвори в потрібних місцях .

Рис.3.13 -збірка Bogie частини шасі

Дальше щоб завершити ланку треба зібрати під ланки коліс з сервоприводом і без .

З ланка з сервоприводом має особливість в приєднанні кронштейна до деталей з’єднання колеса ,це з’єднання виконане в вигляді висі болта з головкою під ключ який обертається на двох підшипниках один кінець вісі з’єднаний з важелем сервопривода а другий з’єднання колеса №2 через само блокуючу гайку, фрагмент на якому зображено це креслення на рисунку 3.14

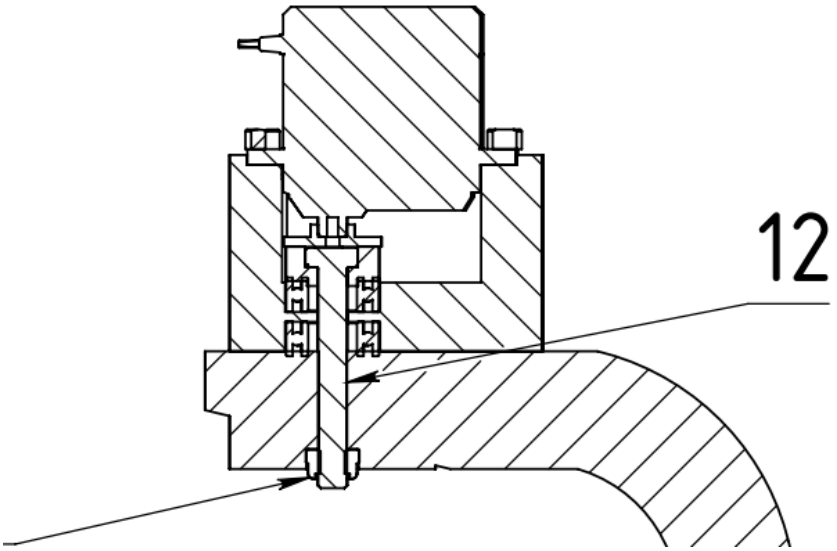


Рисунок 3.14 – з’єднання кронштейна сервопривода з колесом

Саме колесо з’єднуються через муфту яка встановлюється на ва електродвигуна. На рисунку 3.15 показано як обід колеса з’єднається з муфтою через 4 болти М4 і 4 самоблокуючі гайки ,звернути увагу треба на те що гайки розміщенні в протилежний бік до болтів з розвернуті лицем до двигуна .

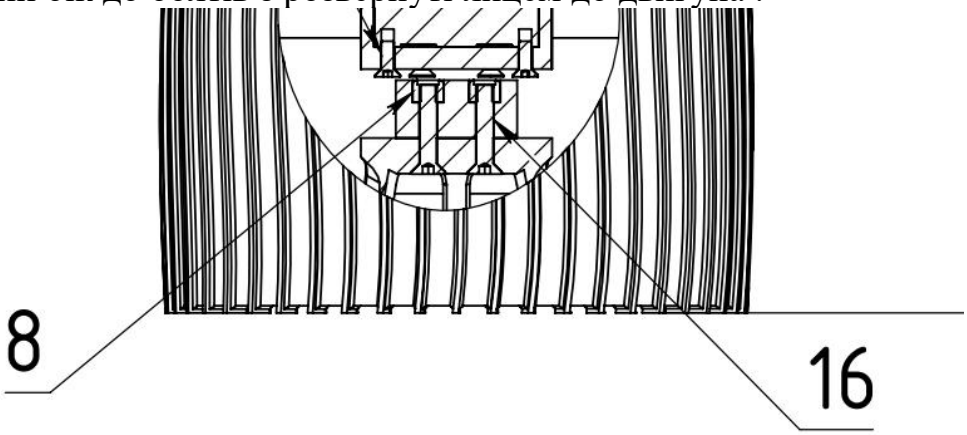


Рисунок 3.15 – з’єднання муфти з обідом колеса

Тепер ми можемо скласти половину ланки шасі .На цьому етапі треба правильно закріпити Bogie до решти Rocker ,щоб підвіска працювала плавно і мала достатню міцність. На рисунку 3.16 показане це з’єднання ,як видно з рисунку ця вісь з’єднання через болт і підшипники щоб зменшити тертя використовуються підшипники

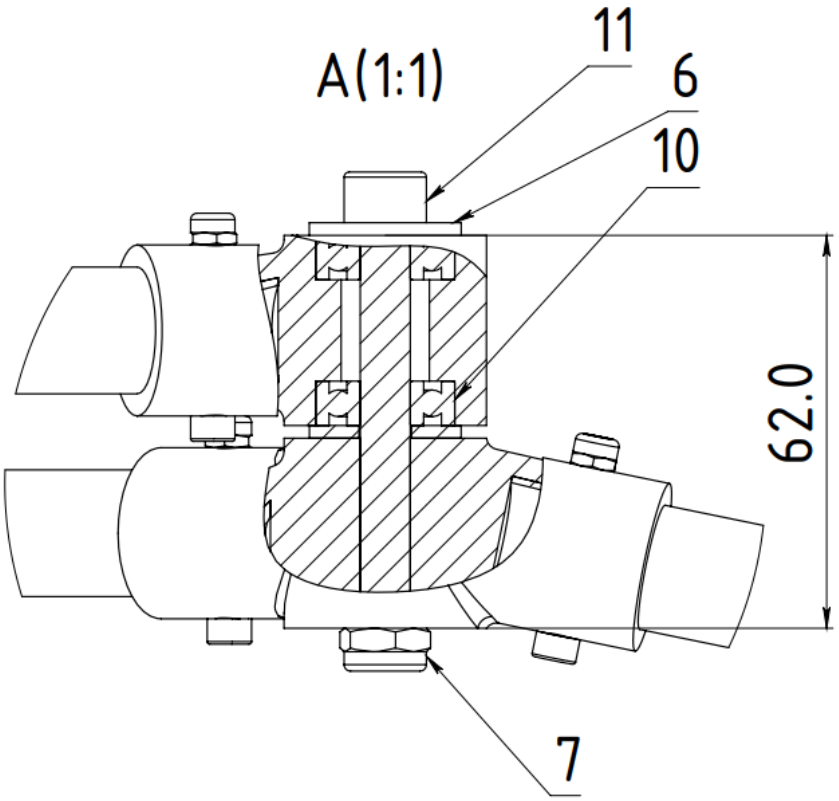


Рисунок 3.16 з’єднання шасі

Рисунок 3.17 Перша збірка шасі

Дальше розберемо основні моменти збірки рами. Спочатку треб зібрати верхню і нижні основи рами за допомогою кутників-кришок які були описані раніше ,за допомогою болтів і т-подібних гайок для верстатних профілів .

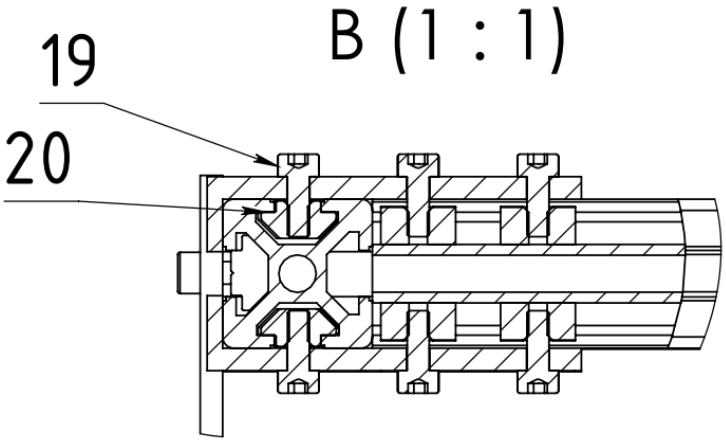


Рисунок 3.18 – кріплення кутників до профілів

Тепер треба посадити диференціальний шарнір на раму ,він закріплюється за допомогою рухомої вісі ,як показано на рисунку 3.19.

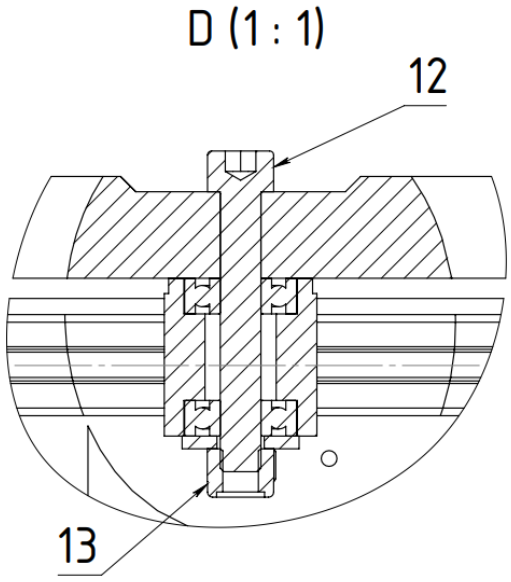


Рисунок 3.19 – з’єднання диференціального шарніру

Диференціальний шарнір з’єднаний з шасі через шарнір кульковий наконечник який вже приєднається через різьбовий стержень який фіксується болтами стандарту ISO 4032.

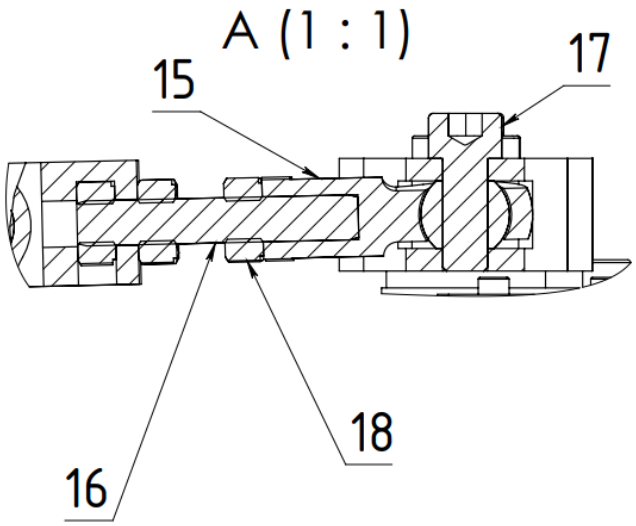


Рисунок 3.20 – зелення диференціального шарніру з шасі

Саме шасі кріпиться до рами через шарнір хитуна ,сам хитун має в собі впаяні латунні втулки М4 стандарту BN1052 щоб забезпечити надійне різьбове з’єднання.Хитун з рамою з’єднаний подібним шляхом як і інші рухомі з’єднання через підшипникову опору ,на рисунку 3.21 зображене це з’єднання.

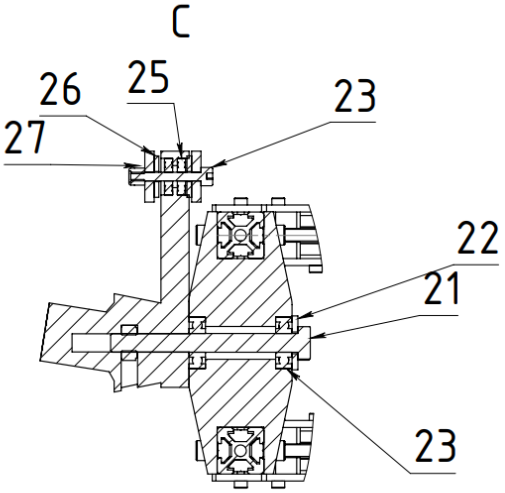


Рисунок 3.21 – з’єднання з хитуна з рамою

Деякі моменти я пропускав бо вони прості і очевидні з креслень які зображені в додатках і не потребують уваги .Тепер можна перейти до монтажу електронної частини марсоходу.

Почнемо з того все живлення і логічні сигнали під’єднанні до блоку керування як показано на рисунках 3.22 і 3.23 .

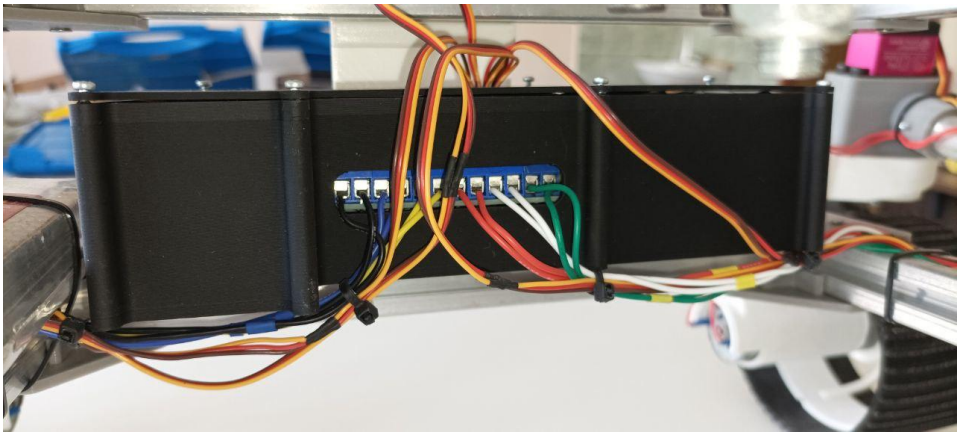


Рисунок 3.22 – передній вихід проводки

Рисунок 3.23 - Верхній вихід проводки

Проводка реалізоване проводами в силіконовій ізоляції ,також треба з’єднувати джгути стяжками щоб проводка прилягали до марсохода .

Виконавши всі монтажні роботи результат збірки можна поглянути на рисунках 3.25..3.28.

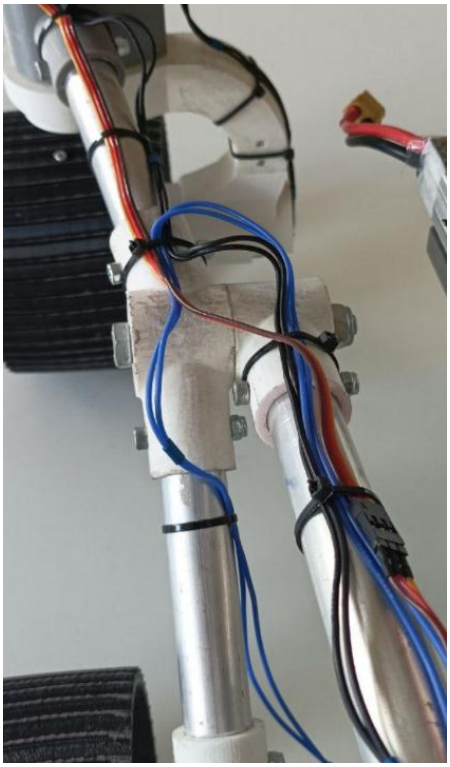


Рисунок 3.24 – монтаж проводки на шасі

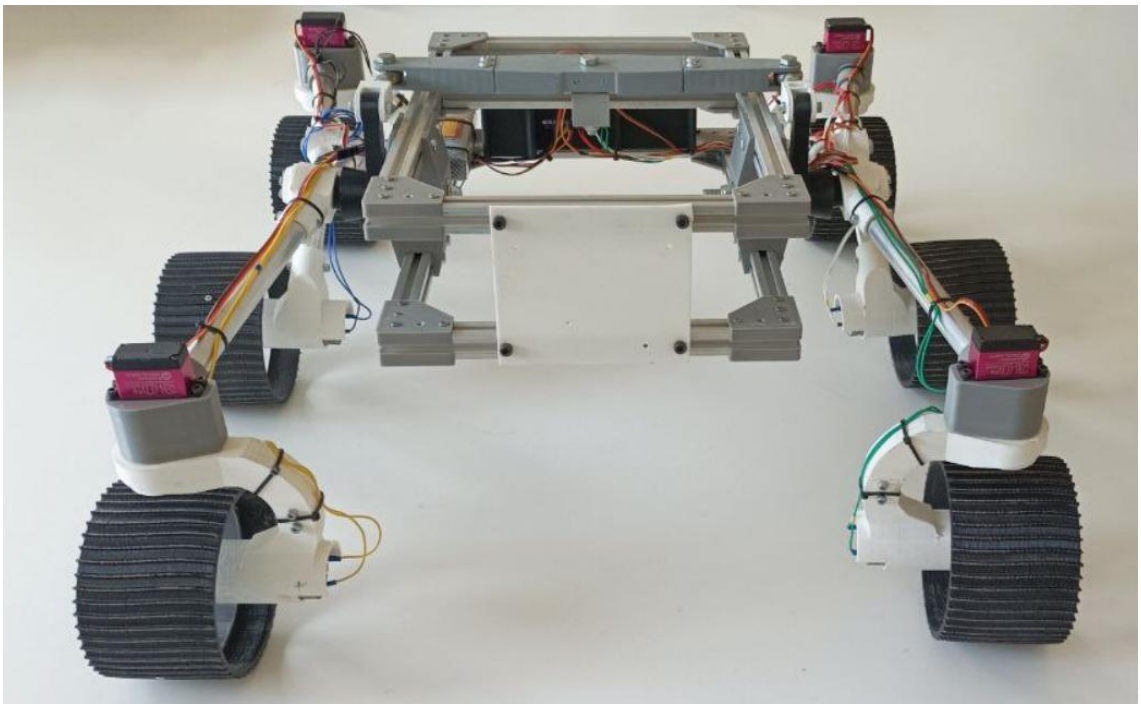


Рисунок 3.25 – модель марсоходу

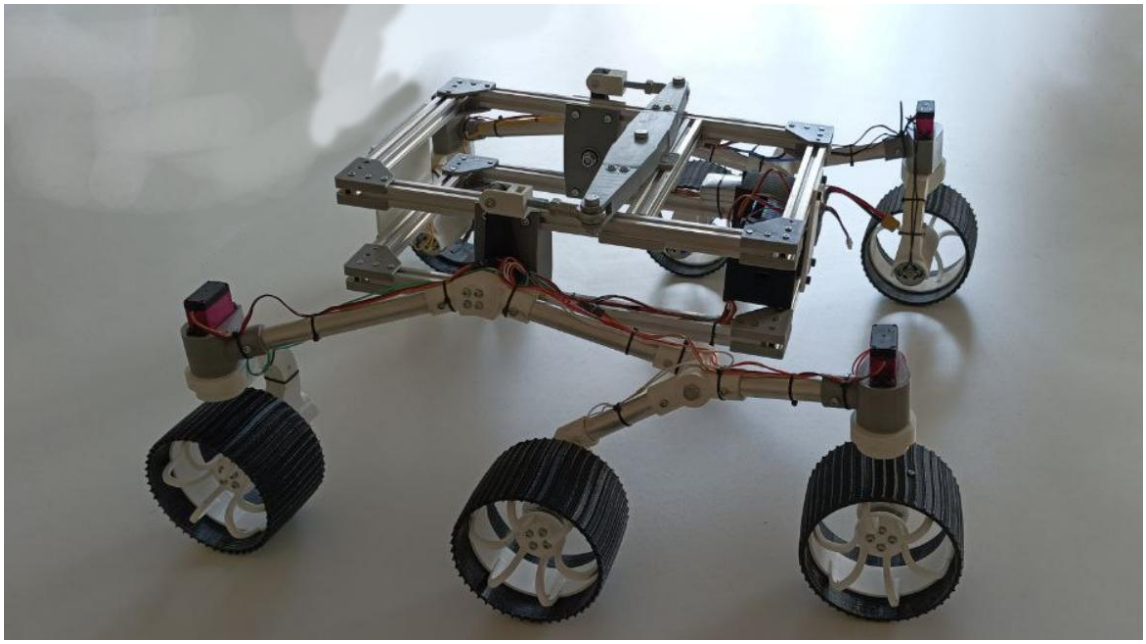


Рисунок 3.26 – модель марсоходу

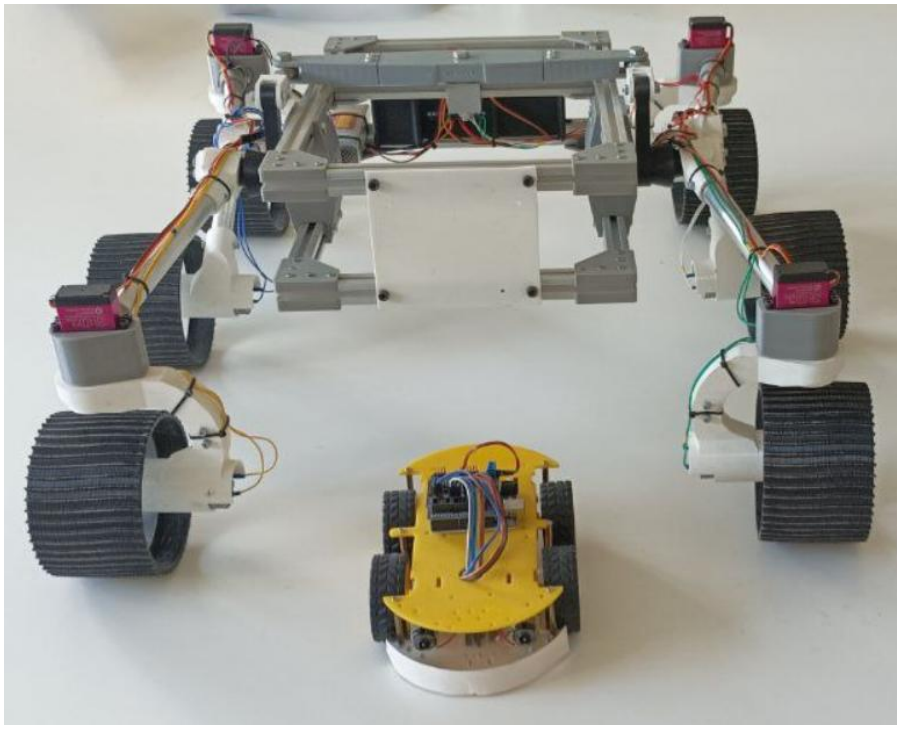


Рисунок 3.27 – порівняння габаритів ”стандартих” роботів

ВИСНОВКИ

У ході виконання роботи було досягнуто поставлену мету — розроблено відносно просту ,доступну ,функціональну і модульну модель марсохода для навчальних цілей.

На основі аналізу літературних джерел ,відкритих технічних матеріалів та документацій марсоходу Perseverence було детально розглянуто особливості будови конструкції .Було детально розглянуто такі елементи як : підвіска типу Rocker-bogie ,конструкцію коліс ,рама та несучі елементи .

Було проведене модернізацію базової моделі ,врахувавши її слабкі місця і усунення їх з створення або модернізації вже наявних деталей .Щоб забезпечити надійність і жорсткість необхідну для цієї конструкції .

Також було проведено поліпшення блоку керування і програмного забезпечення для підвищення його модульності .І відповідно було переписано програмне забезпечення. Після цієї модернізації підвищена загальна обчислювальна здатність ,можливість керування через Wifi ,зменшено використання виводів з мікроконтролера для модальної модернізації марсоходу .

Було підготовленно всі STL-файли для друку деталей ,обранні відповідні матеріали і параметри для кожної деталі . Обрані стандарті компоненти ,для подальшої збірки механічної і електричної частини.

Описонно основні етапи збірки з увагою на нюансами кресленням для наглядності для заміни комплектуючих в разі необхідності модернізації або ремонту.

Практичне значення цієї модель завдяки своїй модульності має великий потенціал для подальших розробок таких як : автоматична навігація ,сенсорні системи для збору даних ,рука маніпулятор і тому подібне.

Такими чином ,результат пророблені роботи показує потенціал цієї моделі для навчальних цілей і показує її як інструмент технічної освіти і розвитку технічно практичних навичок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Science NASA mission [Електронний ресурс] // Mars-2020 Perseverance rover . URL: https://science.nasa.gov/mission/mars-2020-perseverance/
2. NASA mars [Електронний ресурс] //Mars2020 Fact sheet URL: https://mars.nasa.gov/files/mars2020/Mars2020_Fact_Sheet.pdf
3. NASA mission [Електронний ресурс] // Devil's in Details in Selfie Taken by NASA's Mars Perseverance Rover.–URL:

https://www.nasa.gov/missions/mars-2020-perseverance/perseverance-rover/devils-in-details-in-selfie-taken-by-nasas-mars-perseverance-rover/

4. Space.com [Графічний матеріал].// Perseverance rover: Everything you need to know URL: https://www.space.com/perseverance-rover-mars-2020-mission
5. Space.com [Графічний матеріал]. // Sojourner: The first successful Mars rover URL: blob:chrome-untrusted://image- magnify/cbd01a2c-4814-4af0-99d4-460da0a90ebc
6. Hackaday [Електронний ресурс]. // Blog by Lewin day : Rocker Bogie Suspension:The Beloved Solution To Extra-Planetary Rovers URL: https://hackaday.com/2023/09/14/rocker-bogie-suspension-the-beloved-solution-to-extra-planetary-rovers/
7. JLP NASA [Графічний ресурс] // Ramp-drive-test-for-curiosity-mars-rover URL: https://www.jpl.nasa.gov/images/pia13383-ramp-drive-test-for-curiosity-mars-rover/
8. NTRS NASA [Електронний ресурс]. // Utilizing 3D-DIC on the Mars 2020 Rover Wheel Assembly: Test-Analysis Correlation URL: https://ntrs.nasa.gov/api/citations/20220015213/downloads/2023%20IEEE%20Aerospace%20Conference-Full%20Paper-20221003-NWG-JCH-SK-MKC-TFJ.pdf
9. Science NASA [Електронний ресурс]. // Mars 2020: Perseverance Rover Rover Components URL: https://science.nasa.gov/mission/mars-2020-perseverance/rover-components/

.

10. Science NASA [Графічний ресурс] // Mars Perseverance Rover, 3D Model URL : https://science.nasa.gov/resource/mars-perseverance-rover-3d-model/

11. How to Mechatronics [Електронний ресурс]. // Project by Dejan DIY Mars Perseverance Rover Replica – Arduino based Project URL : https://howtomechatronics.com/projects/diy-mars-perseverance-rover-replica-with-arduino/

12. Amazon [Графічний ресурс] // ESP32 DEVKIT ESP32-WROOM-32 Development Board URL : https://www.amazon.com/ESP32-WROOM-32-Development-ESP-32S-Bluetooth-Arduino/dp/B084KWNMM4

13. Makerguides [Електронний ресурс]. // ESP32 vs Arduino Speed Comparison URL : https://www.makerguides.com/esp32-vs-arduino-speed-comparison/

14. Adafruit [Графічний ресурс]. //Adafruit PCA9685 16-Channel Servo Driver URL : https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all

15. SparkFun [Графічний ресурс]. //Motor Driver - Dual TB6612FNG URL : https://www.sparkfun.com/sparkfun-motor-driver-dual-tb6612fng-with-headers.html

16. Docs PlatformIO [Електронний ресурс]. //What is PlatformIO? URL : https://docs.platformio.org/en/latest/what-is-platformio.html

17. Oxorona [Електронний ресурс]. //Сервіс Blynk URL : https://oxorona.com/blynk/

18. Rozetka [Графічний ресурс] // 3D-принтер Vernerfab i3 v2.1 URL : https://rozetka.com.ua/ua/265779221/p265779221/

19. Flying Bear ghost [Графічний ресурс] // Ghost 5 Quick Review URL : https://flyingbearghost.com/en/five

20. Kipp [Графічний ресурс] // Rod ends with ball bearing internal thread, DIN ISO 12240-4 URL : https://www.kipp.com/ie/en/Products/Operating-parts-standard-elements/Joints/Rod-ends-with-ball-bearing-internal-thread-DIN-ISO-12240-4.html

ДОДАТКИ

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Документація		
			ПМІ БР.60.00.00.000	Складальне		
				кресленя		
			ПМІ БР.60.00.00.000	Схема електрично		
				принципова		
				Складальні одиниці		
		1	ПМІ БР.60.04.00.000	Рама	1	
		2	ПМІ БР.60.03.00.000	Трьохопорна	2	
				ланка шасі		
		3	ПМІ БР.60.05.00.000	Модуль керування	1	
				Стантартні вироби		
		4		Втулка різьбова	8	
				латунна М4 BN1052		
		5		Болт М4х20	8	
				DIN 7991		
		6		Болт М3х10	2	
				DIN 7991		
		7		Т-подібна гайка М3	2	
				під паз 6 мм профілю		
				20-ї серії		

A

A

26

Зм.	Арк.	№ докум.	Підп.	Дата	ПМІ БР.60.00.00.000		
Розроб.		Дрегало О.Ю.			Модель марсоходу	Літ.	Маса	Масштаб
Перев.		Панчук В.Г.						
Т. контр.		Панчук В.Г.				Аркуш 2	Аркушів 2	
Реценз.						ІНФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.						
Затв.		Панчук В.Г.						

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Документація		
			ПМІ БР.60.03.00.000	Складальне креслення		
				Складальна одиниця		
		1	ПМІ БР.60.03.01.000	Колосе з сервоприводом	2	
		2	ПМІ БР.60.03.02.000	Колеса без сервоприводу	1	
				Деталі		
		3	ПМІ БР.60.03.00.003	Шарнір хитуна	1	
		4	ПМІ БР.60.03.00.004	Шарнір візка №1	1	
		5	ПМІ БР.60.03.00.005	Шарнір візка №2	1	
				Стантартні вироби		
		6		Шайба М8 DIN 9021	2	
		7		Контргайка М8 DIN EN ISO 10511	1	

А

Зм.	Лист	№ докум.	Підп.	Дата	ПМІ БР.60.03.00.000		
Розроб.		Дрегало О.Ю.			Трьохопорна ланка шасі	Літ.	Маса	Масштаб
Перев.		Панчук В.Г.						
Т. контр.		Панчук В.Г.				Аркуш 2	Аркушів 3	
Реценз						ІНФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.						
Затв.		Панчук В.Г.						

Копировал

Формат А4

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Стантартні вироби		
		8		Болт M4X30	8	
				DIN912		
		9		Контргайка M4	8	
				DIN EN ISO 10511		
		10		Підшибник 608RS	2	
		11		Болт M8X70	1	
				DIN 912		
				Матеріали		
		12		Алюмінієва труба	1	
				D= ∅ 20мм L=197мм		
		13		Алюмінієва труба	1	
				D= ∅ 20мм L=162мм		
		14		Алюмінієва труба	1	
				D= ∅ 20мм L=122мм		
		15		Алюмінієва труба	1	
				D= ∅ 20мм L=116мм		

Зм.	Арк.	№ докум.	Підс.	Дата	Трьохопорна ланка шасі	Арк. 3

Формат А4

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Документація		
			ПМІ БР.60.04.00.000	Складальне Креселння		
				Деталі		
		1	ПМІ БР.60.04.00.001	Основа шарніра хитуна	2	
		2	ПМІ БР.60.04.00.002	Шарнір хитуна	2	
		3	ПМІ БР.60.04.00.003	Тяга диференціального шарніра	2	
		4	ПМІ БР.60.04.00.004	Задня опорна пластина рами	1	
		5	ПМІ БР.60.04.00.005	Передня опорна пластина рами	1	
		6	ПМІ БР.60.04.00.006	Кутовий кронштейн№1	14	
		7	ПМІ БР.60.04.00.007	Т-подібний кронштейн	4	
		8	ПМІ БР.60.04.00.008	Кутовий кронштейн№2	2	
		9	ПМІ БР.60.04.00.009	Диференціальний шарнір частина №1	2	
		10	ПМІ БР.60.04.00.010	Диференціальний шарнір частина №2	1	
		11	ПМІ БР.60.04.00.011	Кронштейн для диференціального шарніра	1	

А

ПМІ БР.60.04 .00.000

Зм.	Арк.	№ докум.	Підп.	Дата
Розроб.		Дрегало О.Ю.		
Пров.		Панчук В.Г.		
Т. контр.		Панчук В.Г.		
Реценз.				
Н. контр.		Панчук В.Г.		
Утв.		Панчук В.Г.		

Рама

Літ.	Масса	Масштаб

Аркуш 2	Аркушів 4

ІНФНТУНГ
ПМІ-21-1

Копіював

Формат А4

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Стантартні вироби		
		12		Болт М8х55	1	
				DIN 912		
		13		Контргайка М8	1	
				DIN EN ISO 10512		
		14		Болт М4х20	4	
				DIN 912		
		15		Шарнір кульковий	2	
				наконечник М8		
				ISO 12240-4		
		16		Різьбовий стрижень	2	
				М8 DIN 975/976		
				L=50мм		
		17		Болт М8Х20	2	
				DIN 912		
		18		Гайка М8	8	
				ISO 4032		
		19		Болт М3х8	89	
				DIN 912		
		20		Т-подібна гайка М3	89	
				під паз 6 мм профілю		
				20-ї серії		
		21		Болт М8Х80	2	
				DIN EN 24015		
		22		Шайба 9.3	3	
				DIN 6908		
		23		Підшибник	6	
				DIN 608RS		

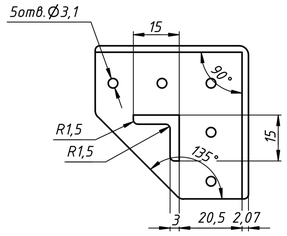
Зм.	Арк.	№ докум.	Підп.	Дата	Рама	Арк. 3

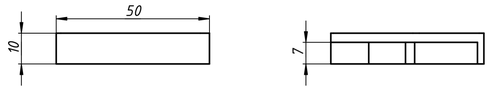
Копіював Формат А4

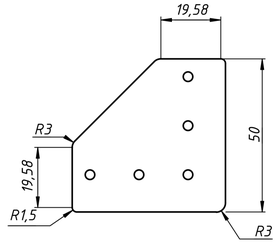
Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Стантартні вироби		
		24		Болт М5х30	2	
				DIN 912		
		25		Підшибник	4	
				DIN 625RS		
		26		Шайба 5.5	4	
				DIN 440-R		
		27		Контргайка М5	2	
				DIN EN ISO 7040		
		28		Болт М5х8	20	
				DIN 912		
		29		Т-подібна гайка М5	20	
				під паз 6 мм профілю		
				20-ї серії		
				Матеріали		
		30		Алюмінієвий	4	
				верстатний профіль		
				20х20 L=283мм		
		31		Алюмінієвий	4	
				верстатний профіль		
				20х20 L=382мм		
		32		Алюмінієвий	1	
				верстатний профіль		
				20х20 L=243мм		

Зм.	Арк.	№ докум.	Підп.	Дата	Рама	Арк. 4

Копіював Формат А4



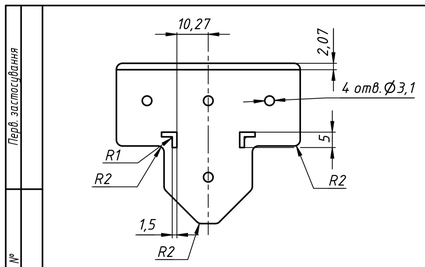


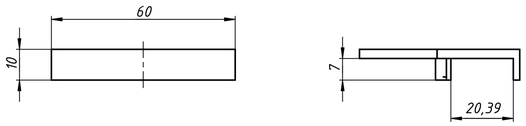


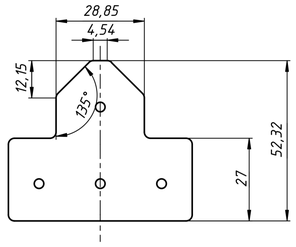
						ПМІ БР.60.04.00.006			
Зм.	Арк.	№ докум.	Підп.	Дата			Літ.	Маса	Масштаб
Розроб.		Дрегало О.Ю.			Кутовий кронштейн№1				1:1
Перев.		Панчук В.Г.					Аркуш 1	Аркушів 1	
Т. контр.		Панчук В.Г.							
Реценз.					Пластик PLA		ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.							
Затв.		Панчук В.Г.							

Перв. застосування | Довідка. № | Підп. і дата | Інв. № дубл. | Зам. інв. № | Підпис і дата | Інв. № од.

Копіював | Формат А4



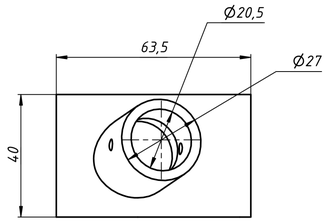


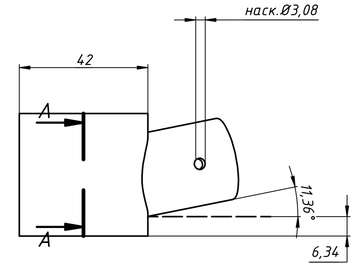


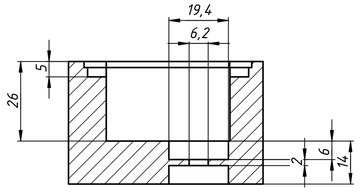
						ПМІ БР.60.04.00.007			
Пом.	Лист	№ докум.	Підп.	Дата			Літ.	Маса	Масштаб
Розроб.		Дрегало О.Ю.			Т-подібний кронштейн				1:1
Пров.		Панчук В.Г.					Аркуш 1	Аркушів 1	
Т. контр.		Панчук В.Г.							
Реценз.					Пластик PLA		ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.							
Затв		Панчук В.Г.							

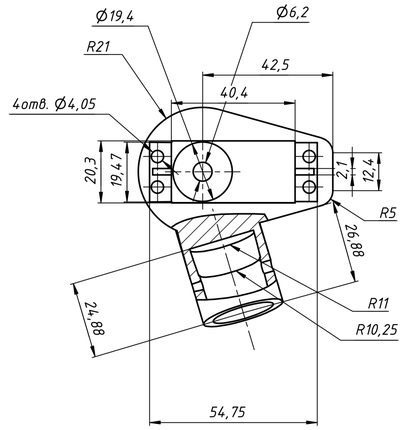
Перв. застосування | Довідка. № | Підп. і дата | Інв. № дубл. | Зам. інв. № | Підпис і дата | Інв. № од.

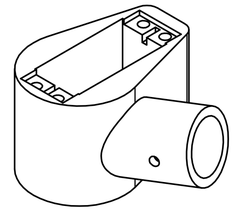
Копіював | Формат А4







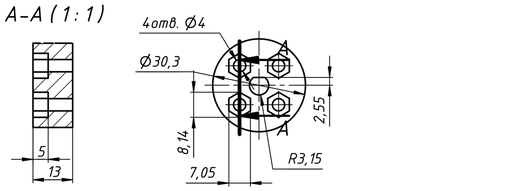


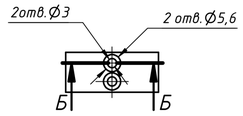


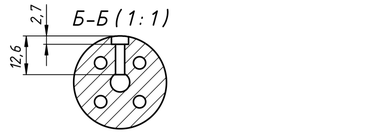
						ПМІ БР.60.03.01.005			
Зм.	Арк.	№ докум.	Підп.	Дата			Літ.	Маса	Масштаб
Розроб.		Дрегало О.Ю.			Кронштейн сервоприводу				
Перев.		Панчук В.Г.					Аркуш 1	Аркушів 1	1
Т. контр.									
Реценз.		Панчук В.Г.			Пластик PLA		ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.							
Затв.		Панчук В.Г.							

Перв. застосування | Довідка. № | Підп. і дата | Інв. № дубл. | Зам. інв. № | Підпис і дата | Інв. № од.

1 Копіював | Формат А3







						ПМІ БР.60.03.01.007			
Зм.	Арк.	№ докум.	Підп.	Дата			Літ.	Маса	Масштаб
Розроб.		Дрегало О.Ю.			Муфта				1:1
Пров.		Панчук В.Г.					Аркуш 1	Аркушів 1	
Т. контр.		Панчук В.Г.							
Реценз.					Пластик PLA		ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.							
Утв.		Панчук В.Г.							

Перв. примірник | Справ. № | Підп. і дата | Інв. № дубл. | Зам. інв. № | Підпис і дата | Інв. № од.

Копіював | Формат А4

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Документація		
			ПМІ БР.60.03.01.000	Складальне креслення		
				Деталі		
		1	ПМІ БР.60.03.01.001	Колесо	1	
		2	ПМІ БР.60.03.01.002	Диск колеса	1	
		3	ПМІ БР.60.03.01.003	З'єднання колеса№1	1	
		4	ПМІ БР.60.03.01.004	З'єднання колеса№2	1	
		5	ПМІ БР.60.03.01.005	Кроштейн сервоприводу	1	
		6	ПМІ БР.60.03.01.006	Важиль сервоприводу	1	
		7	ПМІ БР.60.03.01.007	Муфта	1	
				Стандартні вироби		
		8		Контргайка М4 DIN EN ISO 10511	4	
		9		Болт М3Х8 DIN7991	5	
		10		Підшибник 626RS	2	
		11		Болт М4Х12 DIN912	4	

						ПМІ БР.60.03.01.000			
Зм.	Арк.	№ докум.	Підп.	Дата			Літ.	Маса	Масштаб
Розроб.		Дрегало О.Ю.			Колесо з сервоприводом				
Перев.		Панчук В.Г.					Аркуш 2	Аркушів 3	
Т. контр.		Панчук В.Г.							
Реценз.							ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.							
Затв.		Панчук В.Г.							

1 Копіював | Формат А4

Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Стандартні вироби		
		12		Болт М6Х45 DIN EN 24014	1	
		13		Контр Гайка М6 DIN EN ISO 10511	1	
		14		Болт М3Х10 DIN912	2	
		15		Болт М4Х20 DIN912	2	
		16		Болт М4Х20 DIN7991	4	
				Інші вироби		
		17		Сервопривід DS3225	1	
		18		Двигун DC V=12V d=37mm	1	
		19		Муфта сервоприводу 25T	1	

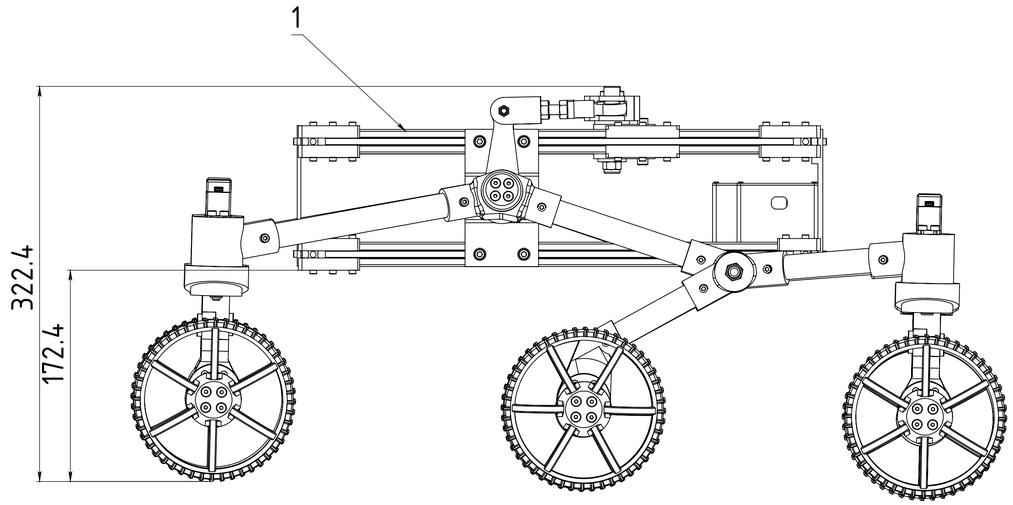
						Колесо з сервоприводом		Арк. 3
Зм.	Арк.	№ докум.	Підп.	Дата				

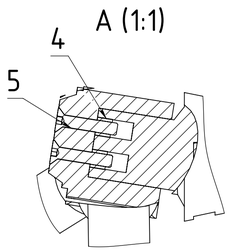
1 Копіював | Формат А4

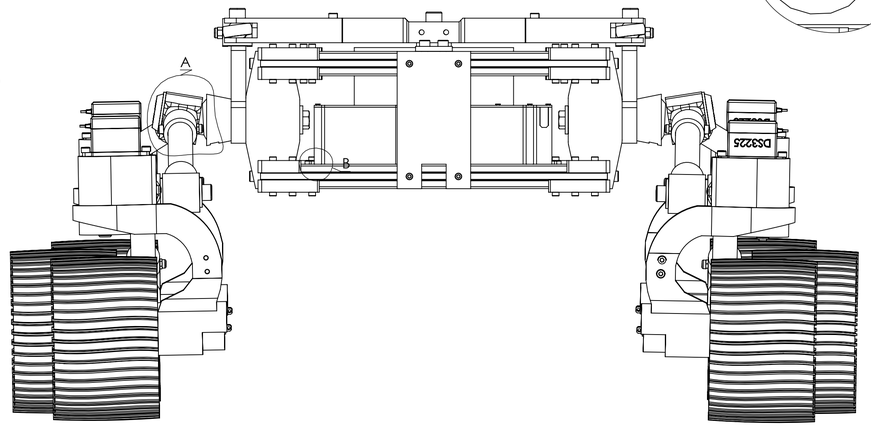
Форма	Зона	Поз.	Позначення	Найменування	Кіл.	Примітка
				Документація		
			ПМІ БР.60.00.000	Схема електрично принципова		
				Деталі		
		U1		ESP32 WROOM-32	1	
		U2		Adafruit 16-канальний 12-бітний ШІМ/Серво Драйвер PCA9685	1	
		U3...U5		Драйвер двигунів двоканальний TB6612FNG	3	
		U6		DC-DC конвертер знижуючий XL4016	1	
		U7		Роз'єм XT60	1	
		X1...X6		Клемники KF301-2P	6	

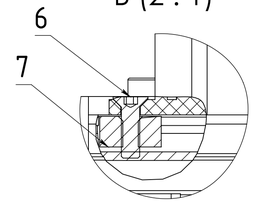
						ПМІ БР.60.00.000			
Зм.	Арк.	№ докум.	Підп.	Дата			Літ.	Маса	Масштаб
Розроб.		Дрегало О.Ю.			Блок керування				
Пров.		Панчук В.Г.					Аркуш 2	Аркушів 2	
Т. контр.		Панчук В.Г.							
Реценз.							ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.							
Утв.		Панчук В.Г.							

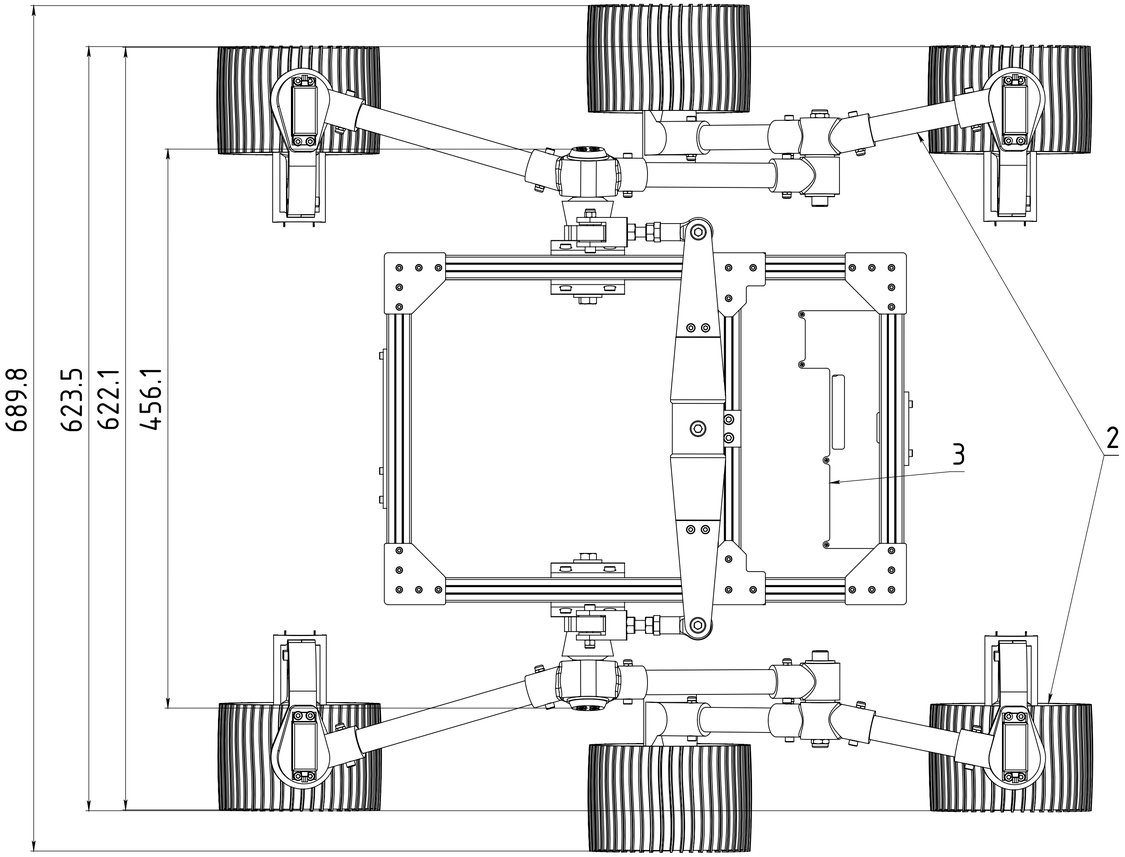
1 Копіював | Формат А4

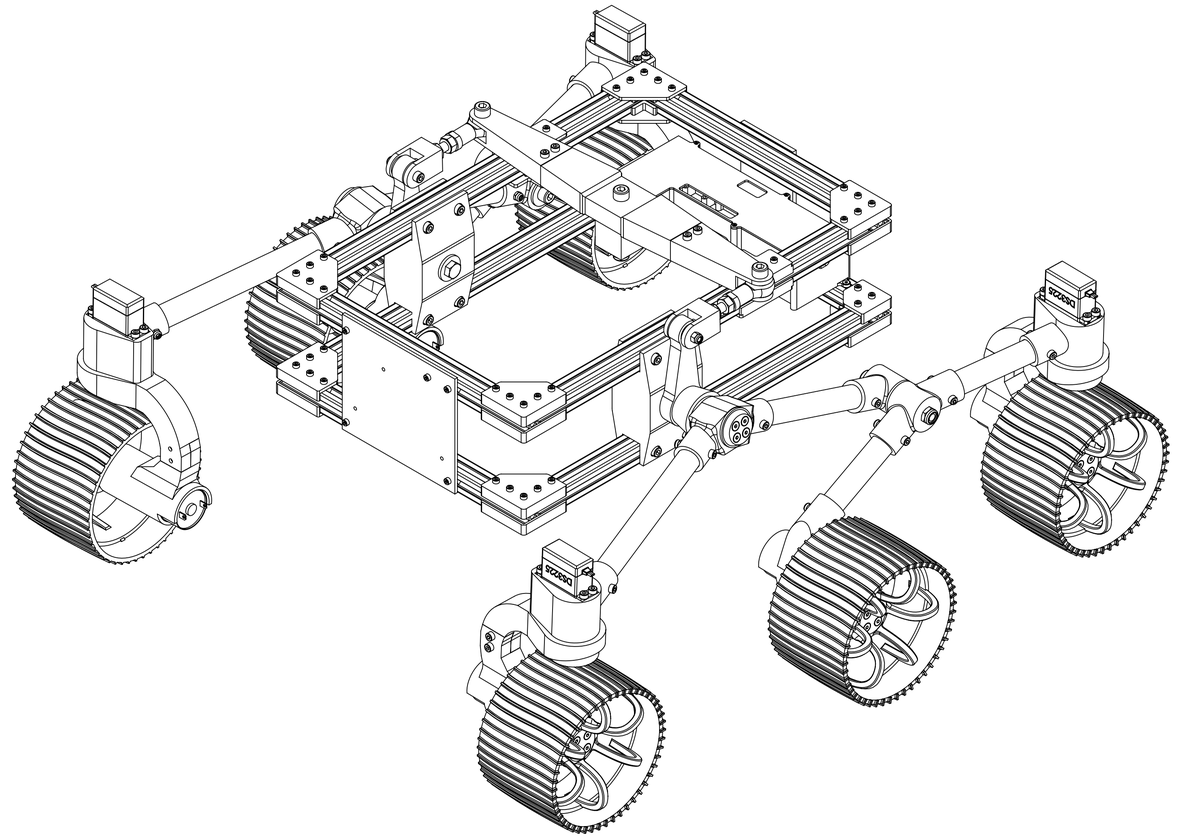










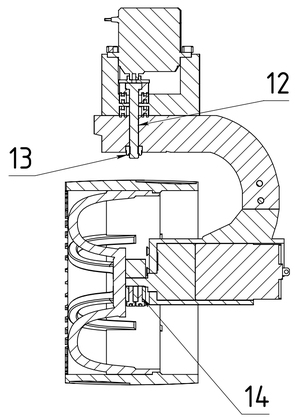


						ПМІ БР.60.00.000 СК	
Зм.	Арк.	№ докум.	Підп.	Дата	Модель марсоходу	Літ.	Масса / Масштаб
Розроб.		Дрегало О.Ю.					1:2,5
Перев.		Панчук В.Г.					
Т. контр.		Панчук В.Г.			Складальне креслення	Аркуш 1	Аркушів 2
Реценз.						ІФНТУНГ ПМІ-21-1	
Н. контр.		Панчук В.Г.					
Затв.		Панчук В.Г.					

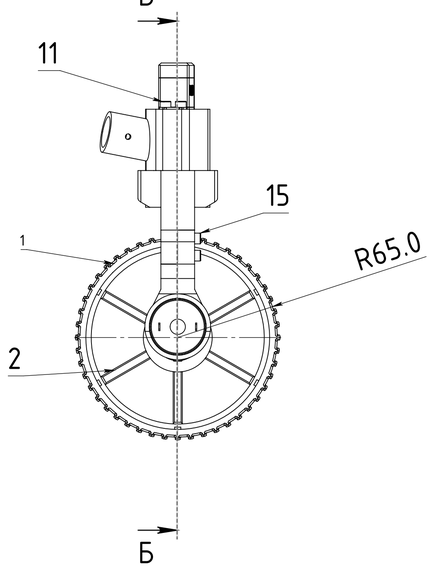
Перв. застосування | Довідковий № | Підп. і дата | Інв. № дубл. | Взаємн. інв. № | Підп. і дата | Інв. № оригін.

Копіював | Формат А1

Б–Б

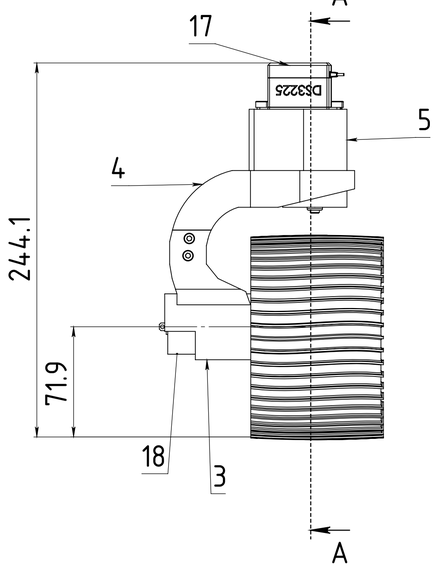


Б



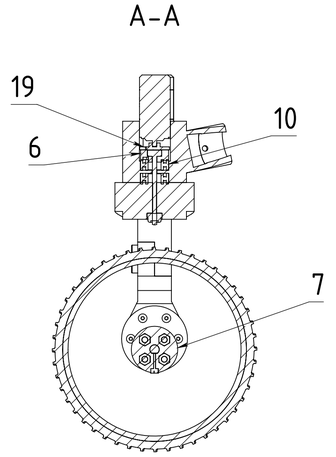
Б

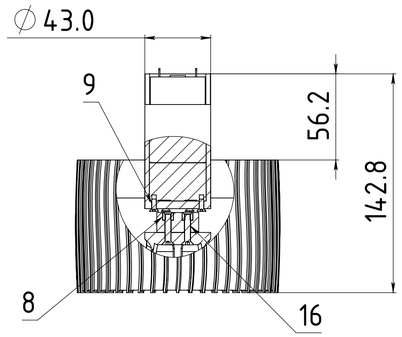
А

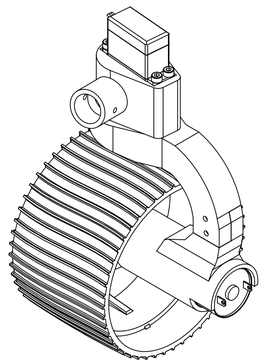


А

А–А



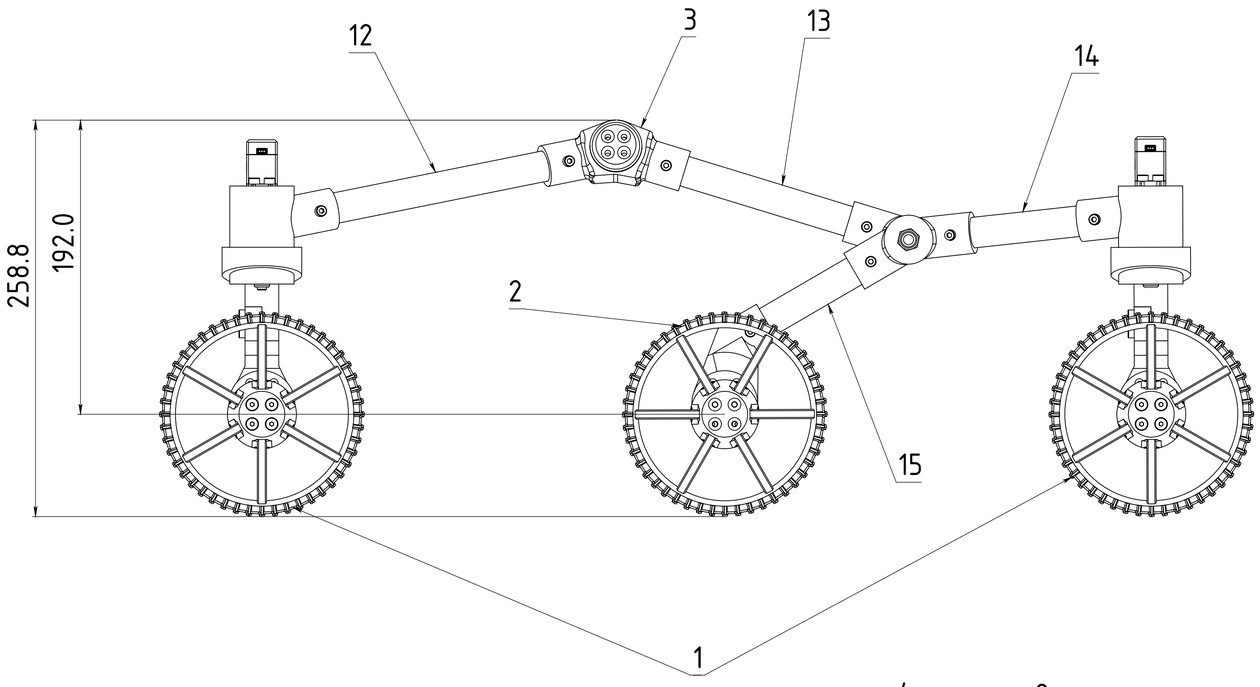


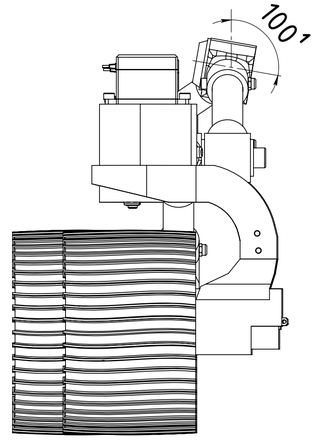


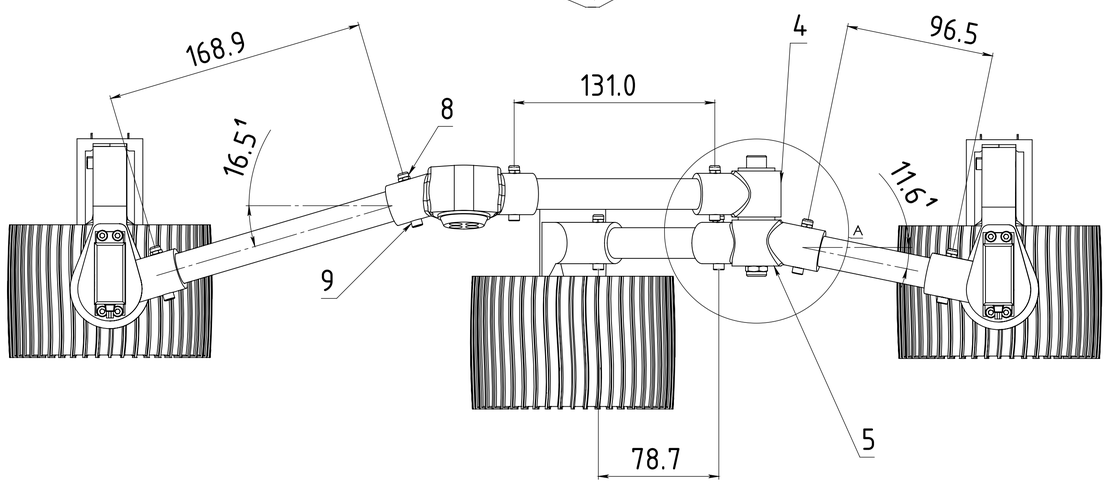
						ПМІ БР.60.03.01.000 СК	
Зм.	Арк.	№ докум.	Підп.	Дата	Колесо з сервоприводом Складальне креслення	Літ.	Масса / Масштаб
Розроб.		Дрегало О.Ю.					1:2
Перев.		Панчук В.Г.				Аркуш 1	Аркушів 3
Т. контр.		Панчук В.Г.				ІФНТУНГ ПМІ–21–1	
Реценз.							
Н. контр.		Панчук В.Г.					
Затв.		Панчук В.Г.					

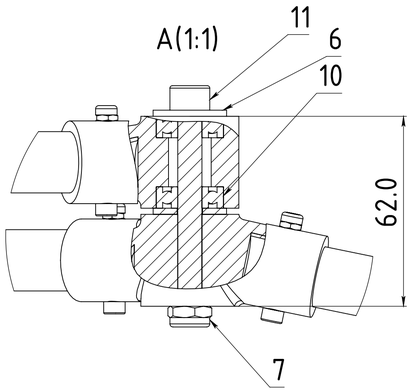
Перед. Застосування. | Довідка. № | Підп. і дата | Інв. № дубл. | Заміст. інв. № | Підп. і дата | Інв. № подл.

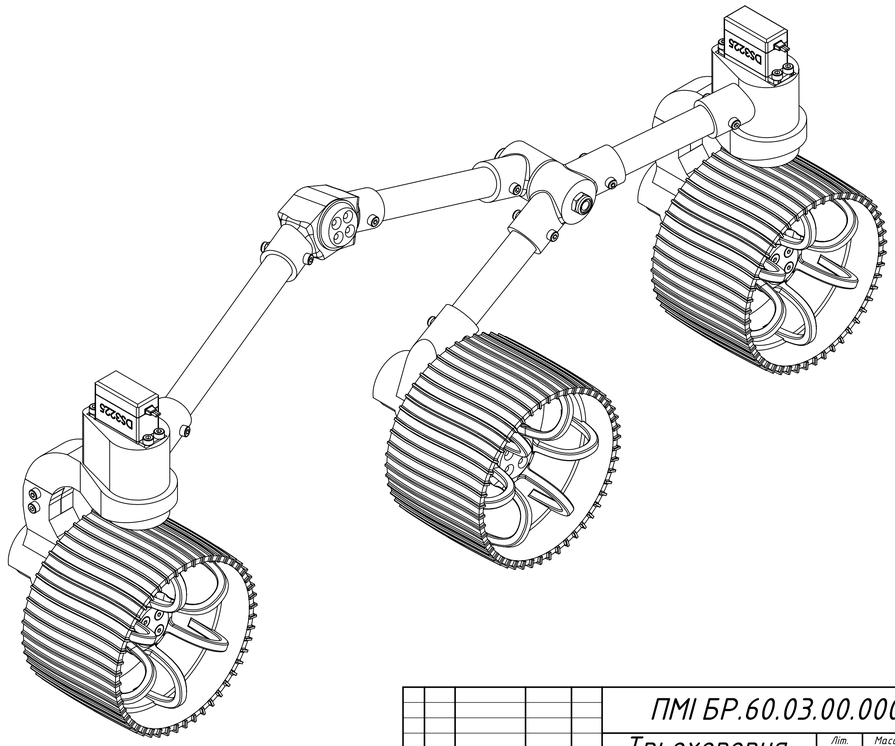
Копіював Формат А1







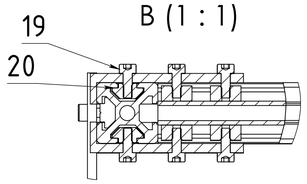


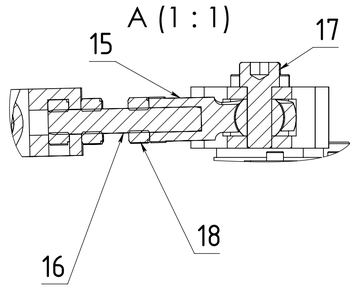


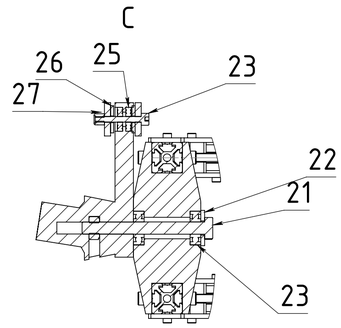
					ПМІ БР.60.03.00.000 СК		
Зм.	Арк.	№ докум.	Підп.	Дата	Трьохопорна ланка шасі	Літ.	Масса / Масштаб
Розроб.		Дрегало О.Ю.					1:2
Перев.		Панчук В.Г.					
Т. контр.		Панчук В.Г.			Складальне креслення	Аркуш 1	Аркушів 3
Реценз.						ІФНТУНГ ПМІ-21-1	
Н. контр.		Панчук В.Г.					
Затв.		Панчук В.Г.					

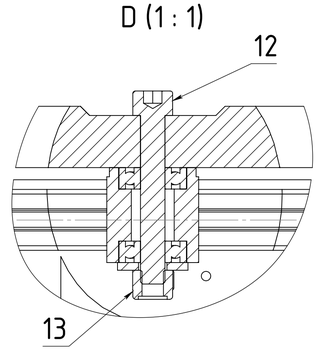
Перед. застосування | Довідка. № | Підп. і дата | Інв. № дубл. | Замість інв. № | Підп. і дата | Інв. № правдн.

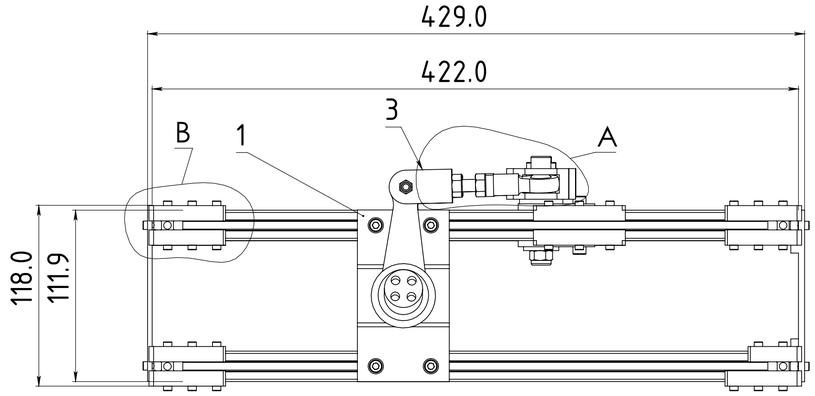
Копіював | Формат А1

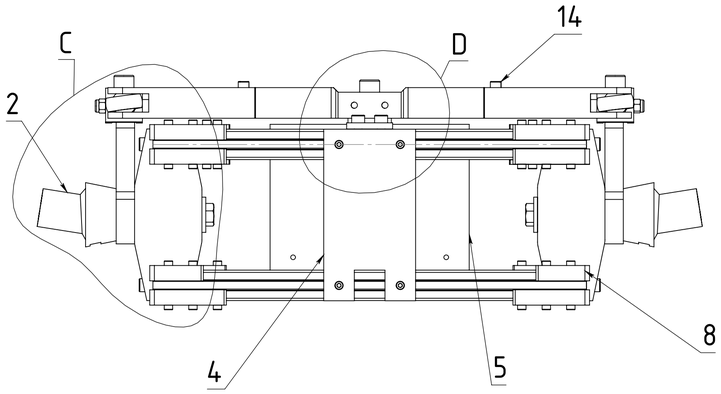


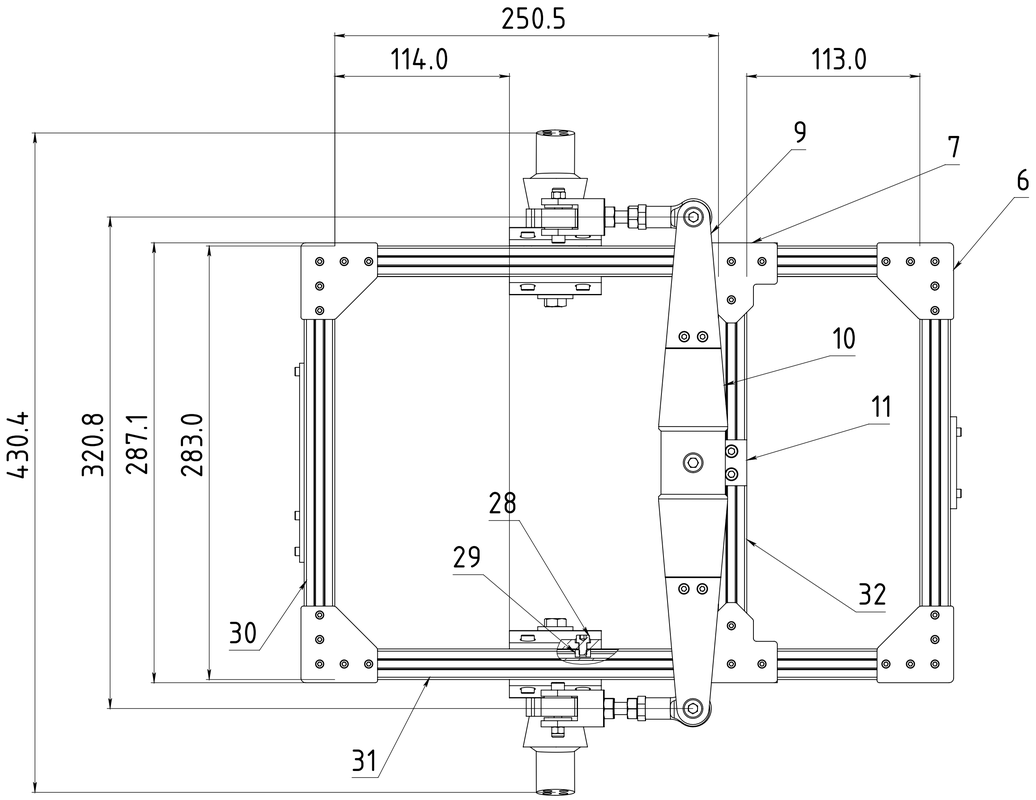


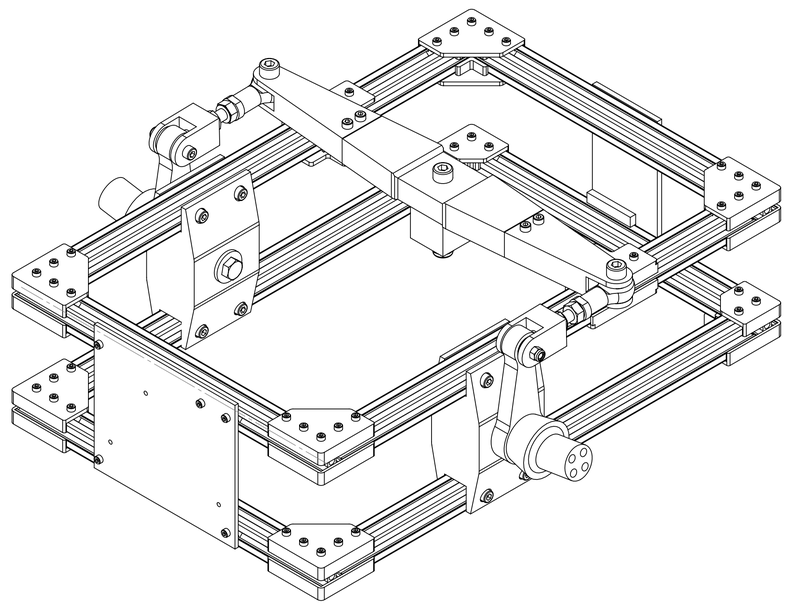








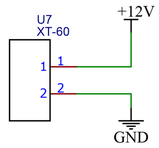


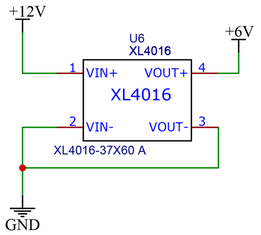


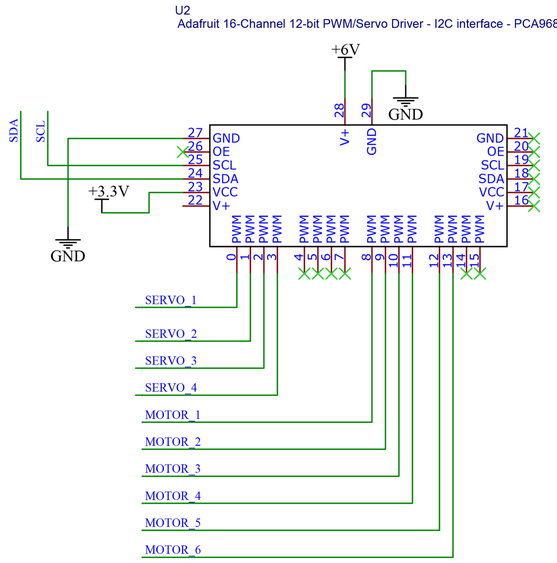
						ПМІ БР.60.04.00.000 СК	
Зм.	Арк.	№ докум.	Підп.	Дата	Рама Складальне креслення	Літ.	Масса	Масштаб
Розроб.		Дрегало О.Ю.						1:2
Перев.		Панчук В.Г.				Аркуш 1	Аркушів 4	
Т. контр.		Панчук В.Г.						
Рецент.						ІФНТУНГ ПМІ-21-1		
Н. контр.		Панчук В.Г.						
Затв.		Панчук В.Г.						

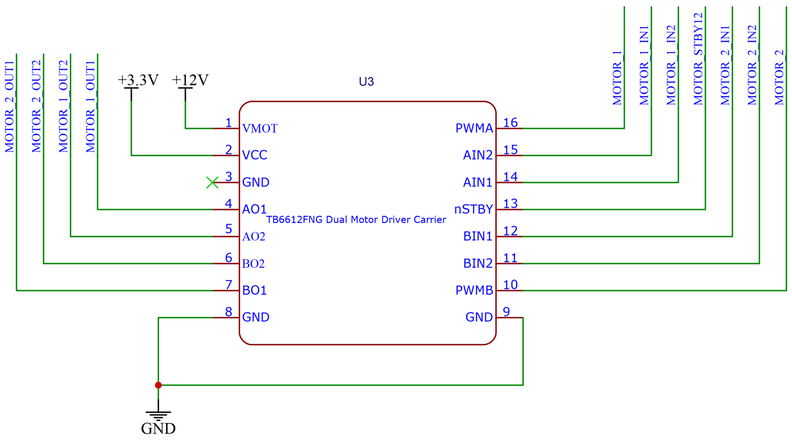
Перв. Застосування
Довідка №
Підп. і дата
Інв. № дубл.
Заміна інв. №
Підп. і дата
Інв. № правдн.

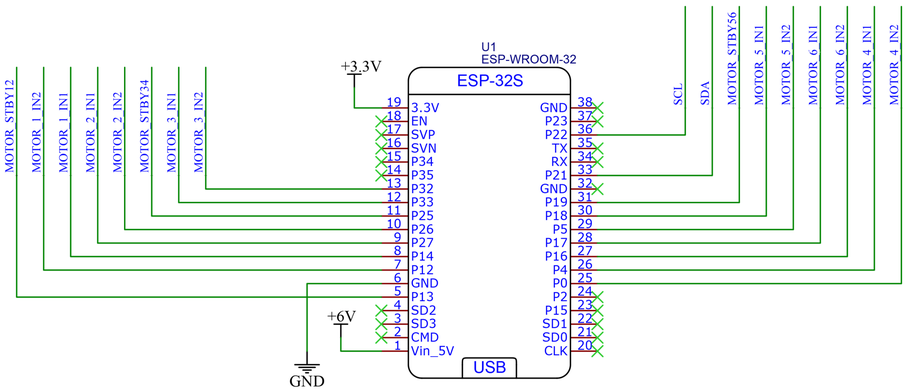
Копіював
Формат А1

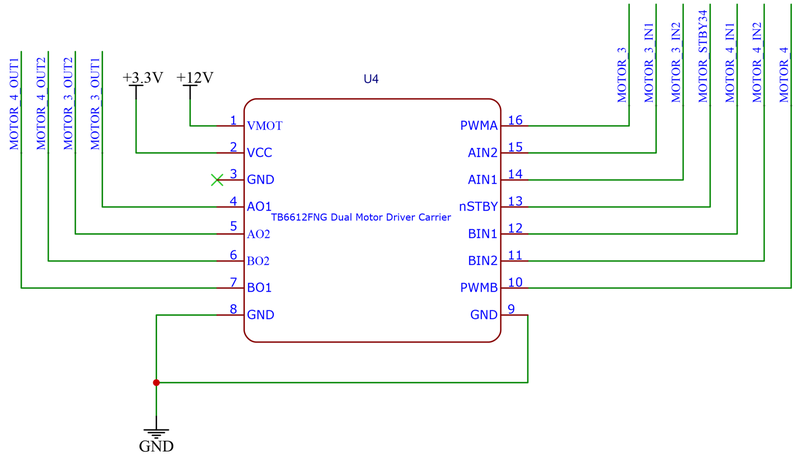


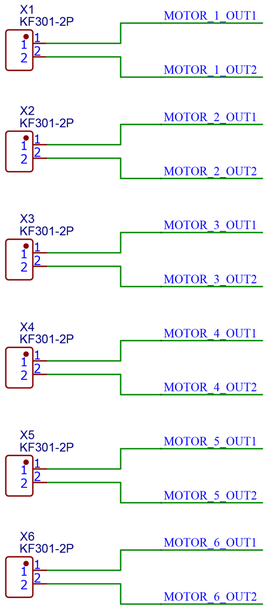


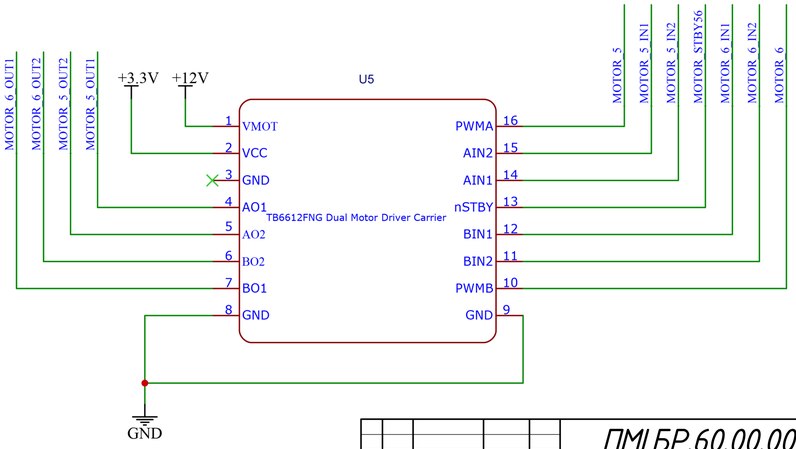












					ПМІ БР.60.00.00.00 СЕП
Зм.	Арк.	№ докум.	Підп.	Дата	Блок Керування
Розроб.		Дрегало О.Ю.			
Перев.		Панчук В.Г.			
Т. контр.		Панчук В.Г.			
Рецент.					
Н. контр.		Панчук В.Г.			
Затв.		Панчук В.Г.			

Літ.	Масса	Масштаб
Аркуш 1	Аркушів 2	
ІФНТУНГ ПМІ-21-1		

Перв. Застосування
Довідка №
Підп. і дата
Інв. № дубл.
Замість інв. №
Підп. і дата
Інв. № правдн.
Копіював
Формат А1

```
#include "config.h"

#include <SPI.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>


// Облікові дані Wi-F
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Mars_rover";
char pass[] = "miniRover";

// Драйвер серво PCA9685
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();


// Ініціалізація основних змінних
int XPlot = 90;
int YPlot = 0;
int YPlot_b = 0;
int servo1Angle = 90;
int servo3Angle = 90;
int servo4Angle = 90;
int servo6Angle = 90;
int speed = 0;
int radius = 0;
float speed1, speed2, speed3 = 0;
float speed1PWM, speed2PWM, speed3PWM = 0;
float thetaInnerFront, thetaInnerBack, thetaOuterFront, thetaOuterBack = 0;


// Встановлення ШІМ для серво
void setServoPWMViaPCA9685(uint8_t servoChannel, int microseconds)
{
    uint16_t pulseLength = map(microseconds, 0, 180, SERVO_MIN, SERVO_MAX);
    pwm.setPWM(servoChannel, 0, pulseLength);
}

// Встановлення ШІМ для мотора
void setMotorPWMViaPCA9685(uint8_t servoChannel, int microseconds)
{
    pwm.setPWM(servoChannel, 0, microseconds);
}

// Ініціалізація PCA9685
void PCA9685_init(int servoInitAngle)
{
    // Ініціалізація PCA9685
    if (!pwm.begin())
    {
        Serial.println("Помилка ініціалізації PCA9685!");
        while (true); // Зупинка, якщо ініціалізація не вдалася
    }
    else
    {
        Serial.println("PCA9685 успішно ініціалізовано.");
    }

    pwm.setPWMFreq(50); // Частота для сервоприводів (50 Гц)

    setServoPWMViaPCA9685(SERVO_1, servoInitAngle);
    setServoPWMViaPCA9685(SERVO_3, servoInitAngle);
    setServoPWMViaPCA9685(SERVO_4, servoInitAngle);
    setServoPWMViaPCA9685(SERVO_6, servoInitAngle);
}

// Ініціалізація пінів моторів
void motor_init()
{
    pinMode(MOTOR_STBY12, OUTPUT);
    pinMode(MOTOR_STBY34, OUTPUT);
    pinMode(MOTOR_STBY56, OUTPUT);
    pinMode(MOTOR_1_IN1, OUTPUT);
    pinMode(MOTOR_1_IN2, OUTPUT);
    pinMode(MOTOR_2_IN1, OUTPUT);
    pinMode(MOTOR_2_IN2, OUTPUT);
    pinMode(MOTOR_3_IN1, OUTPUT);
    pinMode(MOTOR_3_IN2, OUTPUT);
    pinMode(MOTOR_4_IN1, OUTPUT);
    pinMode(MOTOR_4_IN2, OUTPUT);
    pinMode(MOTOR_5_IN1, OUTPUT);
    pinMode(MOTOR_5_IN2, OUTPUT);
    pinMode(MOTOR_6_IN1, OUTPUT);
    pinMode(MOTOR_6_IN2, OUTPUT);

    digitalWrite(MOTOR_STBY12, HIGH);
    digitalWrite(MOTOR_STBY34, HIGH);
    digitalWrite(MOTOR_STBY56, HIGH);
    digitalWrite(MOTOR_1_IN1, LOW);
    digitalWrite(MOTOR_1_IN2, LOW);
    digitalWrite(MOTOR_2_IN1, LOW);
    digitalWrite(MOTOR_2_IN2, LOW);
    digitalWrite(MOTOR_3_IN1, LOW);
    digitalWrite(MOTOR_3_IN2, LOW);
    digitalWrite(MOTOR_4_IN1, LOW);
    digitalWrite(MOTOR_4_IN2, LOW);
    digitalWrite(MOTOR_5_IN1, LOW);
    digitalWrite(MOTOR_5_IN2, LOW);
    digitalWrite(MOTOR_6_IN1, LOW);
    digitalWrite(MOTOR_6_IN2, LOW);
}

void setup()
{
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);

    PCA9685_init(SERVO_INIT_ANGLE);

    motor_init();
}

//Функція виклику Blynk V1
BLYNK_WRITE(V1)
{
    XPlot = param.asInt(); // Отримати значення кута з повзунка (0-180)
}

//Функція виклику Blynk V2
BLYNK_WRITE(V2)
{
    YPlot_b = param.asInt(); // Отримати значення кута з повзунка (-90-90)
}


// Обчислити кут для кожного сервоприводу на основі заданого радіуса повороту
void calculateServoAngle(int radius)
{
    thetaInnerFront = round((atan((LINK_D3 / (radius + LINK_D1)))) * 180 / PI);
    thetaInnerBack = round((atan((LINK_D2 / (radius + LINK_D1)))) * 180 / PI);
    thetaOuterFront = round((atan((LINK_D3 / (radius - LINK_D1)))) * 180 / PI);
    thetaOuterBack = round((atan((LINK_D2 / (radius - LINK_D1)))) * 180 / PI);
}

// Обчислення швидкості двигунів
void calculateMotorsSpeed(int radius, int speed)
{
    // Якщо немає кермового впливу, швидкість усіх коліс однакова — прямолінійний рух
    if (YPlot > 89 && YPlot < 91)
    {
        speed1 = speed2 = speed3 = speed;
    }
    // Під час повороту швидкість коліс залежить від значення радіуса повороту
    else
    {
        // Зовнішні колеса, найдальші від центру повороту, мають найбільшу швидкість
        // Через геометрію ровера всі три зовнішні колеса повинні обертатися майже з однаковою швидкістю.
        // Різниця становить лише близько 1%, тому вважаємо їх однаковими.
        speed1 = speed;
        // Внутрішні передні та задні колеса ближче до точки повороту і мають нижчу швидкість порівняно із зовнішніми колесами
        speed2 = speed * sqrt(pow(LINK_D3, 2) + pow((radius - LINK_D1), 2)) / (radius + LINK_D4);
        // Внутрішнє середнє колесо найближче до точки повороту і має найнижчу швидкість
        speed3 = speed * (radius - LINK_D4) / (radius + LINK_D4);
    }

    // Значення швидкості від 0 до 100% конвертується у PWM від 0 до 4095
    speed1PWM = map(round(speed1), 0, 100, 0, 4095);
    speed2PWM = map(round(speed2), 0, 100, 0, 4095);
    speed3PWM = map(round(speed3), 0, 100, 0, 4095);
}


// Виводимо значення які отримали від Blynk
void printBlynkInput()
{
    Serial.print("YPlot = ");
    Serial.print(YPlot);
    Serial.print(";  XPlot = ");
    Serial.println(XPlot);
}

//Виводимо значення змінних швидкості
void printSpeedInfo()
{
    Serial.print("speed:");
    Serial.println(speed);
    Serial.print("speed1:");
    Serial.println(speed1);
    Serial.print("speed2:");
    Serial.println(speed2);
    Serial.print("speed3:");
    Serial.println(speed3);
    Serial.print("speed1PWM:");
    Serial.println(speed1PWM);
    Serial.print("speed2PWM:");
    Serial.println(speed2PWM);
    Serial.print("speed3PWM:");
    Serial.println(speed3PWM);
}

//Виводимо значення змінних радіусу
void printRadiusInfo()
{
    Serial.print("radius:");
    Serial.println(radius);
    Serial.print("thetaInnerFront:");
    Serial.println(thetaInnerFront);
    Serial.print("thetaInnerBack:");
    Serial.println(thetaInnerBack);
    Serial.print("thetaOuterFront:");
    Serial.println(thetaOuterFront);
    Serial.print("thetaOuterBack:");
    Serial.println(thetaOuterBack);
}

//Рух вправо вперед
void moveRightForward()
{
    // Моторне колесо 1 — переднє ліве
    setServoPWMViaPCA9685(MOTOR_1, speed1PWM); // Зовнішні колеса рухаються зі швидкістю speed1 — максимальною
    digitalWrite(MOTOR_1_IN1, HIGH);
    digitalWrite(MOTOR_1_IN2, LOW);

    // Моторне колесо 2 — ліве середнє
    setServoPWMViaPCA9685(MOTOR_2, speed1PWM);
    digitalWrite(MOTOR_2_IN1, HIGH);
    digitalWrite(MOTOR_2_IN2, LOW);

    // Моторне колесо 3 — ліве заднє
    setServoPWMViaPCA9685(MOTOR_3, speed1PWM);
    digitalWrite(MOTOR_3_IN1, HIGH);
    digitalWrite(MOTOR_3_IN2, LOW);

    // Мотори правої сторони рухаються в протилежному напрямку
    // Моторне колесо 4 — праве передне
    setServoPWMViaPCA9685(MOTOR_4, speed2PWM); // Внутрішнє переднє колесо рухається зі швидкістю speed2 — нижчою швидкістю
    digitalWrite(MOTOR_4_IN1, HIGH);
    digitalWrite(MOTOR_4_IN2, LOW);

    // Моторне колесо 5 — праве середнє
    setServoPWMViaPCA9685(MOTOR_5, speed3PWM); // Внутрішнє середнє колесо рухається зі швидкістю speed3 — мінімальною швидкістю
    digitalWrite(MOTOR_5_IN1, HIGH);
    digitalWrite(MOTOR_5_IN2, LOW);

    // Моторне колесо 6 — праве заднє
    setServoPWMViaPCA9685(MOTOR_6, speed2PWM); // Внутрішнє заднє колесо рухається зі швидкістю speed2 — нижчою швидкістю
    digitalWrite(MOTOR_6_IN1, HIGH);
    digitalWrite(MOTOR_6_IN2, LOW);
}

//Рух вправо назад
void moveRightBack()
{
    // Моторне колесо 1 — ліве передне
    setServoPWMViaPCA9685(MOTOR_1, speed1PWM); // Зовнішні колеса рухаються зі швидкістю speed1 — максимальною
    digitalWrite(MOTOR_1_IN1, LOW);
    digitalWrite(MOTOR_1_IN2, HIGH);

    // Моторне колесо 2 — ліве середнє
    setServoPWMViaPCA9685(MOTOR_2, speed1PWM);
    digitalWrite(MOTOR_2_IN1, LOW);
    digitalWrite(MOTOR_2_IN2, HIGH);

    // Моторне колесо 3 — ліве заднє
    setServoPWMViaPCA9685(MOTOR_3, speed1PWM);
    digitalWrite(MOTOR_3_IN1, LOW);
    digitalWrite(MOTOR_3_IN2, HIGH);

    // Мотори правої сторони рухаються в протилежному напрямку
    // Моторне колесо 4 — праве передне
    setServoPWMViaPCA9685(MOTOR_4, speed2PWM); // Внутрішнє переднє колесо рухається зі швидкістю speed2 — нижчою швидкістю
    digitalWrite(MOTOR_4_IN1, LOW);
    digitalWrite(MOTOR_4_IN2, HIGH);

    // Моторне колесо 5 — праве середнє
    setServoPWMViaPCA9685(MOTOR_5, speed3PWM); // Внутрішнє середнє колесо рухається зі швидкістю speed3 — найнижчою швидкістю
    digitalWrite(MOTOR_5_IN1, HIGH);
    digitalWrite(MOTOR_5_IN2, LOW);

    // Моторне колесо 6 — праве заднє
    setServoPWMViaPCA9685(MOTOR_6, speed2PWM); // Внутрішнє заднє колесо рухається зі швидкістю speed2 — нижчою швидкістю
    digitalWrite(MOTOR_6_IN1, LOW);
    digitalWrite(MOTOR_6_IN2, HIGH);
}

// Рух вліво вперед
void moveLeftForward()
{
    // Моторне колесо 1 — ліве передне
    setServoPWMViaPCA9685(MOTOR_1, speed2PWM);
    digitalWrite(MOTOR_1_IN1, HIGH);
    digitalWrite(MOTOR_1_IN2, LOW);

    // Моторне колесо 2 — ліве середнє
    setServoPWMViaPCA9685(MOTOR_2, speed3PWM);
    digitalWrite(MOTOR_2_IN1, HIGH);
    digitalWrite(MOTOR_2_IN2, LOW);

    // Моторне колесо 3 — ліве заднє
    setServoPWMViaPCA9685(MOTOR_3, speed2PWM);
    digitalWrite(MOTOR_3_IN1, HIGH);
    digitalWrite(MOTOR_3_IN2, LOW);

    // Мотори правої сторони рухаються в протилежному напрямку
    // Моторне колесо 4 — праве передне
    setServoPWMViaPCA9685(MOTOR_4, speed1PWM);
    digitalWrite(MOTOR_4_IN1, HIGH);
    digitalWrite(MOTOR_4_IN2, LOW);

    // Моторне колесо 5 — праве середнє
    setServoPWMViaPCA9685(MOTOR_5, speed1PWM);
    digitalWrite(MOTOR_5_IN1, LOW);
    digitalWrite(MOTOR_5_IN2, HIGH);

    // Моторне колесо 6 — праве заднє
    setServoPWMViaPCA9685(MOTOR_6, speed1PWM);
    digitalWrite(MOTOR_6_IN1, LOW);
    digitalWrite(MOTOR_6_IN2, HIGH);
}

// Рух вліво назад
void moveLeftBack()
{
    // Моторне колесо 1 — ліве передне
    setServoPWMViaPCA9685(MOTOR_1, speed2PWM);
    digitalWrite(MOTOR_1_IN1, LOW);
    digitalWrite(MOTOR_1_IN2, HIGH);

    // Моторне колесо 2 — ліве середнє
    setServoPWMViaPCA9685(MOTOR_2, speed3PWM);
    digitalWrite(MOTOR_2_IN1, LOW);
    digitalWrite(MOTOR_2_IN2, HIGH);

    // Моторне колесо 3 — ліве заднє
    setServoPWMViaPCA9685(MOTOR_3, speed2PWM);
    digitalWrite(MOTOR_3_IN1, LOW);
    digitalWrite(MOTOR_3_IN2, HIGH);

    // Мотори правої сторони рухаються в протилежному напрямку
    // Моторне колесо 4 — праве передне
    setServoPWMViaPCA9685(MOTOR_4, speed1PWM);
    digitalWrite(MOTOR_4_IN1, LOW);
    digitalWrite(MOTOR_4_IN2, HIGH);

    // Моторне колесо 5 — праве середнє
    setServoPWMViaPCA9685(MOTOR_5, speed1PWM);
    digitalWrite(MOTOR_5_IN1, LOW);
    digitalWrite(MOTOR_5_IN2, HIGH);

    // Моторне колесо 6 — праве заднє
    setServoPWMViaPCA9685(MOTOR_6, speed1PWM);
    digitalWrite(MOTOR_6_IN1, LOW);
    digitalWrite(MOTOR_6_IN2, HIGH);
}

// Рух прямо вперед
void moveStraightForward()
{
    // Моторне колесо 1 — ліве передне
```

Зм.	Арк.	№ докум.	Підп.	Дата
Розроб.		Дрегало О.Ю.		
Перев.		Панчук В.Г.		
Т. контр.		Панчук В.Г.		
Рецент.				
Н. контр.		Панчук В.Г.		
Затв.		Панчук В.Г.		

ПМІ БР.60.00.00.000 СХ

Програма керуванням марсоходом

Літ.	Маса	Масштаб
Аркуш 1	Аркушів 2	

ІФНТУНГ ПМІ-21-1

Перв. Застосування | Довідка № | Підп. і дата | Інв. № дубл. | Взам. інв. № | Підп. і дата | Інв. № подл.

Копіював Формат А1

```
    setMotorPWMViaPCA9685(MOTOR_1, speed1PWM);
    digitalWrite(MOTOR_1_IN1, HIGH);
    digitalWrite(MOTOR_1_IN2, LOW);

    // Моторне колесо 2 — ліве середнє
    setMotorPWMViaPCA9685(MOTOR_2, speed1PWM);
    digitalWrite(MOTOR_2_IN1, HIGH);
    digitalWrite(MOTOR_2_IN2, LOW);

    // Моторне колесо 3 — ліве заднє
    setMotorPWMViaPCA9685(MOTOR_3, speed1PWM);
    digitalWrite(MOTOR_3_IN1, HIGH);
    digitalWrite(MOTOR_3_IN2, LOW);

    // Мотори правої сторони рухаються в протилежному напрямку
    // Моторне колесо 4 — праве переднє
    setMotorPWMViaPCA9685(MOTOR_4, speed1PWM);
    digitalWrite(MOTOR_4_IN1, HIGH);
    digitalWrite(MOTOR_4_IN2, LOW);

    // Моторне колесо 5 — праве середнє
    setMotorPWMViaPCA9685(MOTOR_5, speed1PWM);
    digitalWrite(MOTOR_5_IN1, HIGH);
    digitalWrite(MOTOR_5_IN2, LOW);

    // Моторне колесо 6 — праве заднє
    setMotorPWMViaPCA9685(MOTOR_6, speed1PWM);
    digitalWrite(MOTOR_6_IN1, HIGH);
    digitalWrite(MOTOR_6_IN2, LOW);
}

// Рух вперед назад
void moveStraightBack()
{
    // Motor Wheel 1 - Left Front
    setMotorPWMViaPCA9685(MOTOR_1, speed1PWM);
    digitalWrite(MOTOR_1_IN1, LOW);
    digitalWrite(MOTOR_1_IN2, HIGH);

    // Motor Wheel 2 - Left Middle
    setMotorPWMViaPCA9685(MOTOR_2, speed1PWM);
    digitalWrite(MOTOR_2_IN1, LOW);
    digitalWrite(MOTOR_2_IN2, HIGH);

    // Motor Wheel 3 - Left Back
    setMotorPWMViaPCA9685(MOTOR_3, speed1PWM);
    digitalWrite(MOTOR_3_IN1, LOW);
    digitalWrite(MOTOR_3_IN2, HIGH);

    // right side motors move in opposite direction
    // Motor Wheel 4 - Right Front
    setMotorPWMViaPCA9685(MOTOR_4, speed1PWM);
    digitalWrite(MOTOR_4_IN1, LOW);
    digitalWrite(MOTOR_4_IN2, HIGH);

    // Motor Wheel 5 - Right Middle
    setMotorPWMViaPCA9685(MOTOR_5, speed1PWM);
    digitalWrite(MOTOR_5_IN1, LOW);
    digitalWrite(MOTOR_5_IN2, HIGH);

    // Motor Wheel 6 - Right Back
    setMotorPWMViaPCA9685(MOTOR_6, speed1PWM);
    digitalWrite(MOTOR_6_IN1, LOW);
    digitalWrite(MOTOR_6_IN2, HIGH);
}


//зупинка ровера
void motorStop()
{
    digitalWrite(MOTOR_1_IN1, LOW);
    digitalWrite(MOTOR_1_IN2, LOW);
    digitalWrite(MOTOR_2_IN1, LOW);
    digitalWrite(MOTOR_2_IN2, LOW);
    digitalWrite(MOTOR_3_IN1, LOW);
    digitalWrite(MOTOR_3_IN2, LOW);
    digitalWrite(MOTOR_4_IN1, LOW);
    digitalWrite(MOTOR_4_IN2, LOW);
    digitalWrite(MOTOR_5_IN1, LOW);
    digitalWrite(MOTOR_5_IN2, LOW);
    digitalWrite(MOTOR_6_IN1, LOW);
    digitalWrite(MOTOR_6_IN2, LOW);
}

void loop()
{
    Blynk.run();
    if (YPlot_b < 0) YPlot = YPlot_b * -1;
    else YPlot = YPlot_b;


#ifdef DEBUG_PRINT
    printBlynkInput();
#endif

    if (XPlot > 90)
    {
        radius = map(XPlot, 90, 180, 1400, 600); // радіус повороту від 1400 мм до 600 мм
    }
    // Поворот вліво
    else if (XPlot < 90)
    {
        radius = map(XPlot, 90, 0, 1400, 600); // радіус повороту від 1400 мм до 600 мм
    }
    speed = map(YPlot, 0, 90, 0, 100); // швидкість ровера від 0% до 100%


    calculateMotorsSpeed(radius, speed);
    calculateServoAngle(radius);

#ifdef DEBUG_PRINT
    printSpeedInfo();
    printRadiusInfo();
#endif

    // Поворот вправо
    if (XPlot > 90)
    {
        // Серводвигуни
        // Зовнішні колеса
        setServoPWMViaPCA9685(SERVO_1, servo1Angle + thetaInnerFront); // переднє колесо повертає вправо
        setServoPWMViaPCA9685(SERVO_3, servo3Angle - thetaInnerBack);
        // заднє колесо повертає вліво для загального повороту вправо ровера
        // Внутрішні колеса
        setServoPWMViaPCA9685(SERVO_4, servo4Angle + thetaOuterFront);
        setServoPWMViaPCA9685(SERVO_6, servo6Angle - thetaOuterBack);

        if (YPlot_b > 0)
        {
            // Рух вперед
            moveRightForward();
        }
        else if (YPlot_b < 0)
        {
            moveRightBack();
        }
        else motorStop();
    }

    // Поворот вліво
    else if (XPlot < 90)
    {
        setServoPWMViaPCA9685(SERVO_1, servo1Angle - thetaOuterFront);
        setServoPWMViaPCA9685(SERVO_3, servo3Angle + thetaOuterBack);
        setServoPWMViaPCA9685(SERVO_4, servo4Angle - thetaInnerFront);
        setServoPWMViaPCA9685(SERVO_6, servo6Angle + thetaInnerBack);

        if (YPlot_b > 0)
        {
            // Рух вперед
            moveLeftForward();
        }
        else if (YPlot_b < 0)
        {
            moveLeftBack();
        }
        else motorStop();
    }
    // Рух прямо
    else
    {
        setServoPWMViaPCA9685(SERVO_1, servo1Angle);
        setServoPWMViaPCA9685(SERVO_3, servo3Angle);
        setServoPWMViaPCA9685(SERVO_4, servo4Angle);
        setServoPWMViaPCA9685(SERVO_6, servo6Angle);

        if (YPlot_b > 0)
        {
            // Рух вперед
            moveStraightForward();
        }
        else if (YPlot_b < 0)
        {
            moveStraightBack();
        }
        else motorStop();
    }

}
```

config.h

```
#pragma once

#define BLYNK_TEMPLATE_ID "TMPL4JI3RnWx2"
#define BLYNK_TEMPLATE_NAME "Контроллер"
#define BLYNK_AUTH_TOKEN "HDlO1vk34uTedswQ40NmP8UxGOEEK3Z5"

//конфігурування каналів керування
#define SERVO_1 0
#define SERVO_3 1
#define SERVO_4 2
#define SERVO_6 3



#define MOTOR_STBY12 13

#define MOTOR_1 8
#define MOTOR_1_IN1 14
#define MOTOR_1_IN2 12

#define MOTOR_2 9
#define MOTOR_2_IN1 27
#define MOTOR_2_IN2 26

#define MOTOR_STBY34 25

#define MOTOR_3 10
#define MOTOR_3_IN1 33
#define MOTOR_3_IN2 32

#define MOTOR_4 11
#define MOTOR_4_IN1 0
#define MOTOR_4_IN2 4

#define MOTOR_STBY56 19

#define MOTOR_5 12
#define MOTOR_5_IN1 18
#define MOTOR_5_IN2 5

#define MOTOR_6 13
#define MOTOR_6_IN1 17
#define MOTOR_6_IN2 16

#define BLYNK_PRINT Serial

#define SERVO_MIN 80    // Мінімальний сигнал для сервоприводу (~0 градусів)
#define SERVO_MAX 550   // Максимальний сигнал для сервоприводу (~180 градусів)
#define SERVO_INIT_ANGLE 90

#define LINK_D1 271.0f
#define LINK_D2 278.0f
#define LINK_D3 301.0f
#define LINK_D4 304.0f

#define DEBUG_PRINT
```

Підп. і дата | Інв. № дубл. | Замість інв. № | Підп. і дата | Інв. № підл.

Зм.	Арк.	№ докум.	Підп	Дата	Програма керування марсоходом	Арк. 2

Копіював | Формат А1