Term Project

MORPHOBOT

Department of Mechanical and Aerospace Engineering,

New York University Tandon School of Engineering,

Brooklyn, NY

ROB-GY 6103 | Spring 2024

05/14/2024

Harshavardhan Vibhandik (hsv2015)
Dinesh Sai Naidu Rallapalli (dr3696)
Satya Krishna Kotla (sk10849)
Sanket Patil (ssp6669)

Table of Contents

Abstract

—

This project focuses on creating Morphobot, a versatile robot that can move on the ground and fly. It uses a Raspberry Pi 5 to control its movements on land and an Arduino for flying, making it easy to switch between rolling and flying with just a button press. Morphobot is equipped with motor drivers for driving and sensors like gyroscopes and accelerometers for stable flying, allowing it to handle different kinds of terrain.

A special feature of Morphobot is its ability to change how it moves based on where it is and what it encounters. Users can control it manually or set it to explore on its own, gathering data as it goes. This makes Morphobot a great tool for exploring hard-to-reach places.

This project combines robotics, system design, and smart technology to show how robots can be more adaptable and helpful in various situations. Morphobot is not just about exploring new places; it's also about making sophisticated robots more user-friendly and accessible for everyone.

1. Introduction

Robotics has evolved significantly over the years, with each advancement making robots more capable and easier to use. Our Morphobot project represents the next step in this evolution. It uniquely combines the capabilities of moving on the ground and flying through the air, making it ideal for exploring hard to reach places.

Our goal with Morphobot is to create a robot that can smoothly transition between rolling on the ground and flying, much like switching between apps on a smartphone. This dual-mode functionality is perfect for navigating through complex environments, whether in busy urban settings or rugged natural landscapes.

Thanks to its portability and user-friendliness, people can take the Morphobot with them wherever they go for learning and exploration.

Improving the Morphobot's mobility and accessibility expands the possibilities for learning and exploration rather than just improving a technological component.

2. Key Components

- **Lipo Battery Voltage Checker:** This voltage checker and alarm monitor the state of the Lipo battery during use, alerting us if the voltage drops to a level that could potentially damage the battery or impair the rover's performance. This tool helps prevent over-discharge during operation, which is crucial for preserving the battery's life and ensuring the rover can complete its tasks.

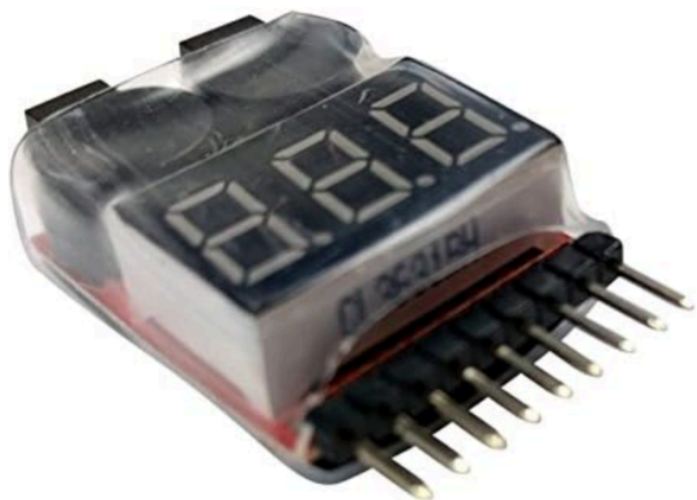


Fig 2.1

- **LiPo Charger:** This Lipo charger is essential for recharging our batteries efficiently and safely, ensuring that the Morphobot is always ready to go. It supports a variety of battery types, including the LiPo batteries we use. The charger also supports balance charging, which helps extend the life and performance of our batteries by evenly charging each cell. This is critical for keeping the batteries healthy and providing constant performance during operations.

Fig 2.2

- **LiPo Battery:** This high-capacity battery powers our Morphobot, providing the necessary energy for both its ground and aerial operations.

Fig 2.3

Its 11.1V output is ideal for supporting the motor and control systems, ensuring that the rover can operate efficiently during extended missions. The Deans T Connector facilitates a secure and reliable connection to the rover's power system, minimizing the risk of disconnections during operation.

- **Motor Drive Controller Board:** The L298N Motor Drive Controller Boards were pivotal in controlling MorphoBot's

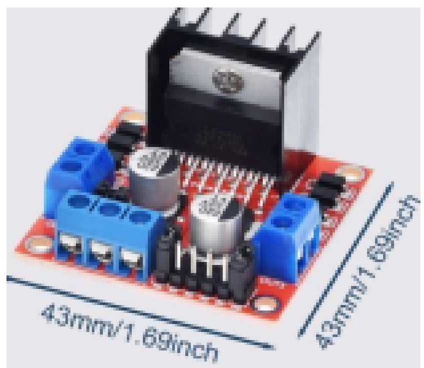


Fig 2.4

movement. By enabling the control of two DC motors simultaneously, they allowed for intricate movements and precise navigation. Their compatibility with Arduino facilitated easy integration into our rover's control system.

- **Arduino:** The Arduino is the flight controller for the Morphobot's aerial capabilities. It handles all aspects of flight control, from keeping the rover stable in the air to performing difficult maneuvers. The Arduino is designed to manage real-time adjustments based on a variety of sensor inputs, allowing the rover to respond rapidly and effectively to changes in its environment. This makes it extremely useful for maintaining control and stability during flight, allowing for precise aerial navigation.

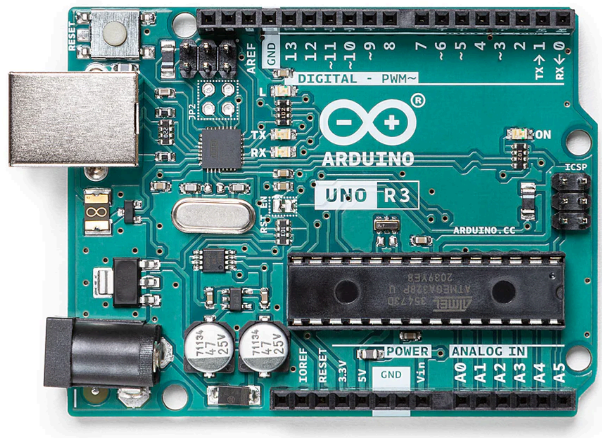


Fig. 2.5

- **Raspberry Pi 5:** The Raspberry Pi 5 is vital for Morphobot because it controls all the ground movements. It uses its powerful processor to run programs that help Morphobot move accurately and smoothly over different types of ground.

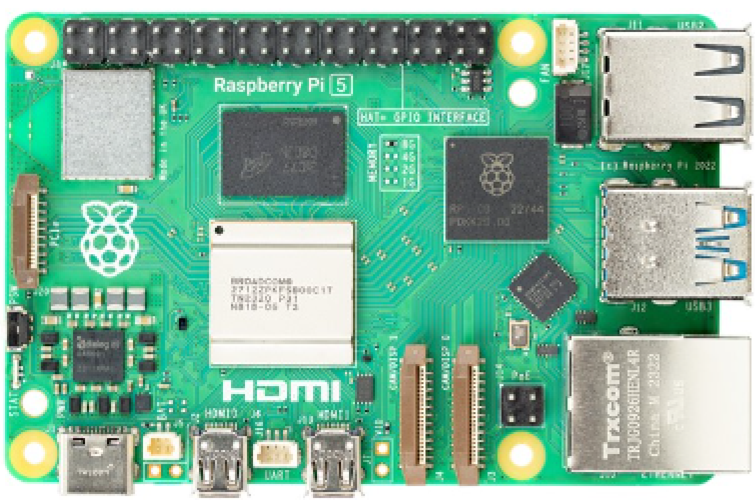


Fig. 2.6

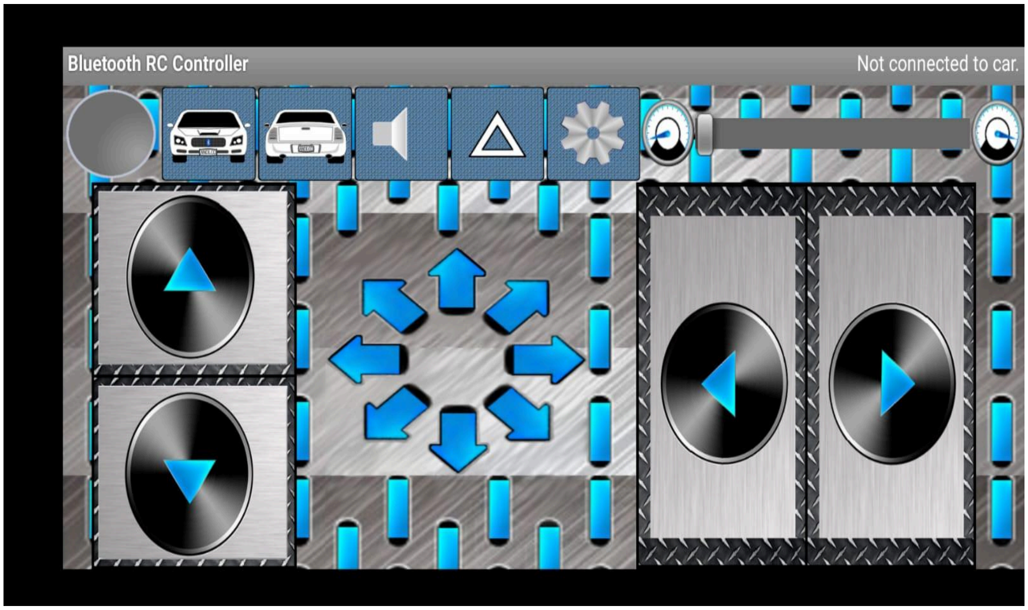
It also allows Morphobot to be controlled from a distance, making it great for tasks like exploring and watching over areas. This makes the Raspberry Pi 5 a key part of how Morphobot works.

Description	Total Cost
Lipo Battery Voltage Checker	$5.49
Lipo Charger	$39.99
LiPo Battery	$11.99
Arduino	$27.60
Motor Drive Controller	$7.98
Raspberry Pi5	$90
Total	***$218.05***

3. Main Features

1. **Dual Mode Functionality (Raspberry Pi 5 & Arduino):** Morphobot is designed with a dual control system that utilizes the strengths of both the Raspberry Pi 5 and Arduino to manage its ground and aerial functions, respectively. This dual-controller setup ensures that Morphobot operates efficiently in its hybrid role, adapting seamlessly to the demands of different environments and tasks.
    - **Raspberry Pi 5: Ground Control:** The Raspberry Pi 5 serves as the brain for ground operations. This capability is essential for maneuvering through challenging or cluttered landscapes, ensuring reliable and autonomous ground movement.
    - **Arduino: Aerial Control:** This setup provides precise and stable control during flight, which is vital for conducting aerial surveys and accessing areas that are otherwise unreachable by ground. The Arduino's ability to manage intricate flight dynamics enhances Morphobot's utility in a wide range of applications, from environmental monitoring to emergency response.

- **Remote Operation (Bluetooth and RF Communication):**
    1. **Bluetooth Module (Raspberry Pi 5):** Allows for wireless control using UART protocol over short distances, making it ideal for operations where the rover needs to stay within the line of sight of the operator. The module is accessed using the mobile application (Bluetooth RC controller). The interface is as follows:



The major controls used are :

- Forward: F
- Backward - B
- Left - L
- Right - R
- Slow speed - W
- Medium speed - U
- High Speed - V
- Stop - S

2. **RF Communication (Arduino):** Enables long-range control, particularly useful for aerial drone operations, where the rover might need to travel beyond visual range. The communication is accessed using the FS-i6X (10 Channel) controller, which communicates serially with the Arduino.

- **Advanced Navigation and Obstacle Detection:**

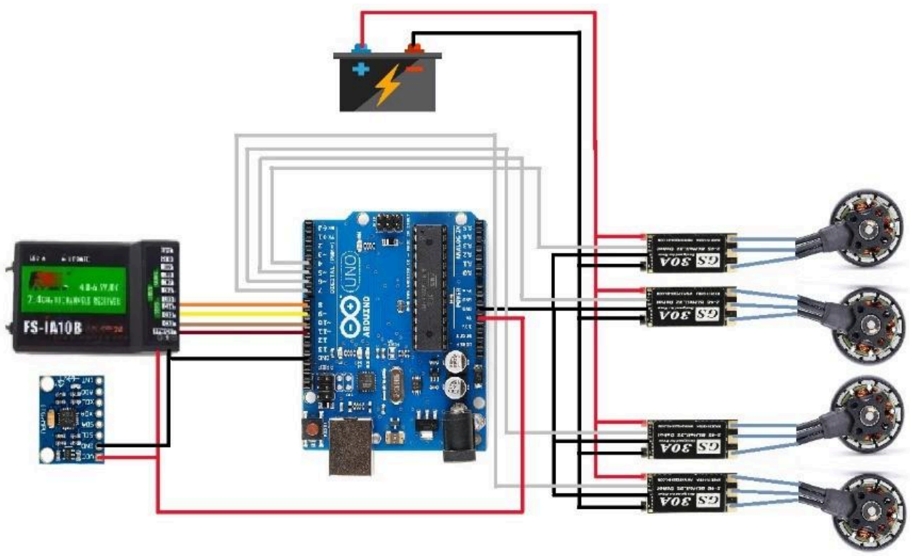
1. **MPU-6050 Gyroscope and Accelerometer (Arduino):** Provide critical data on orientation and acceleration, which help stabilize the rover during flight and improve maneuverability.

- **Power Management and Safety:**
    1. **Zeee 11.1V 50C 3000mAh 3S Lipo Battery:** Powers both the ground and aerial modes of the rover, ensuring long operational periods without the need for frequent recharging.
    2. **Apex RC Products Lipo Battery Voltage Checker Alarm:** Monitors the battery's voltage to prevent damage from over-discharging, enhancing the safety and longevity of the Morphobot.

- **Efficient Power Transfer and Charging:**

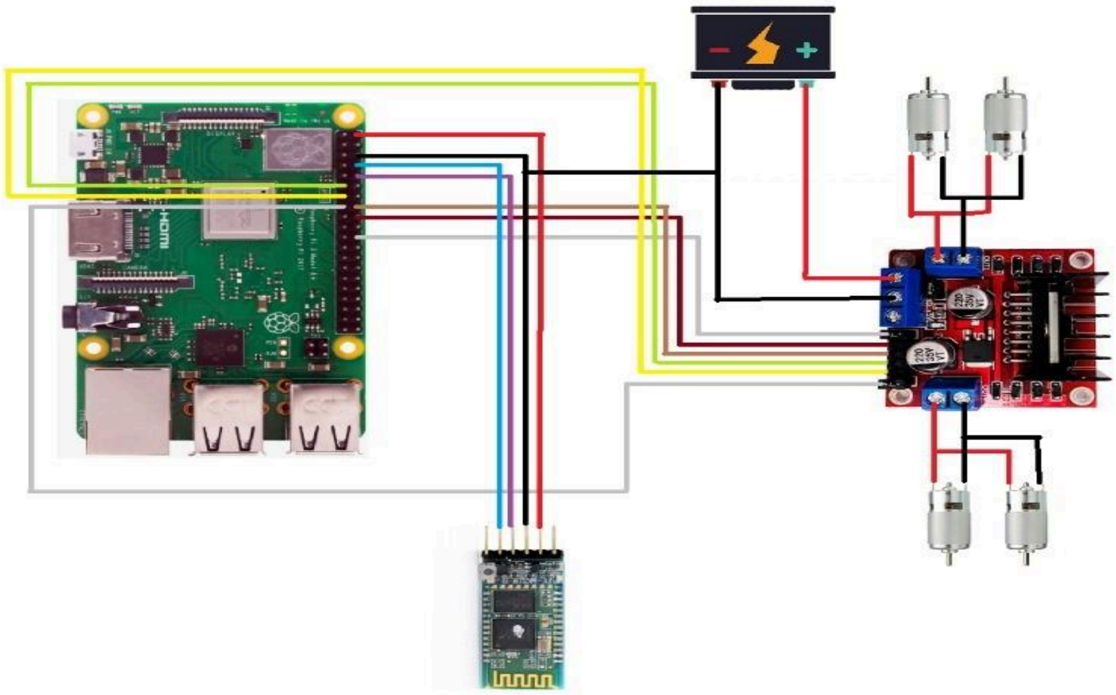
1. **Deans T Connector (on Lipo Battery):** Ensures a secure and efficient power connection, critical for maintaining consistent performance during extensive operations.
2. **Lipo Charger H B6 RC Charger:** Offers efficient and balanced charging, crucial for maintaining battery health and readiness for extended missions.

4. Circuit Design



- **Arduino-Based Circuit Design:** The circuit design for controlling a robot using an Arduino and various components. The Arduino board, which is the main controller, is connected to a motor driver that controls four motors. These motors are likely for moving the robot. There's also a servo controller in the circuit, which is used to manage movements of other mechanical parts, like arms or adjustments in the robot's orientation.

  Additionally, there's a battery connected to provide power to the system, and a Bluetooth module that allows wireless communication. This setup lets you control the robot remotely, probably from a computer or a smartphone, to send commands to drive the motors or adjust the servos. This kind of system is typically used to give the robot the ability to move around and perform tasks autonomously or through remote control.



- **Circuit Design for Raspberry Pi:** It shows a circuit design for a robot that includes several key components to control its operations. The central part of the circuit is a Raspberry Pi, a small computer that acts as the brain of the robot. It's connected to a motor driver, which controls two motors. These motors are likely used for moving the robot. There's also a battery that provides the necessary power for the Raspberry Pi and the motors.

  In addition, there's a Bluetooth module connected to the Raspberry Pi, allowing for wireless communication. This setup lets someone control the robot remotely, sending commands to move or perform other actions. This kind of system is designed to make the robot mobile and capable of performing tasks autonomously or through remote instructions.

5. Mechanical Design

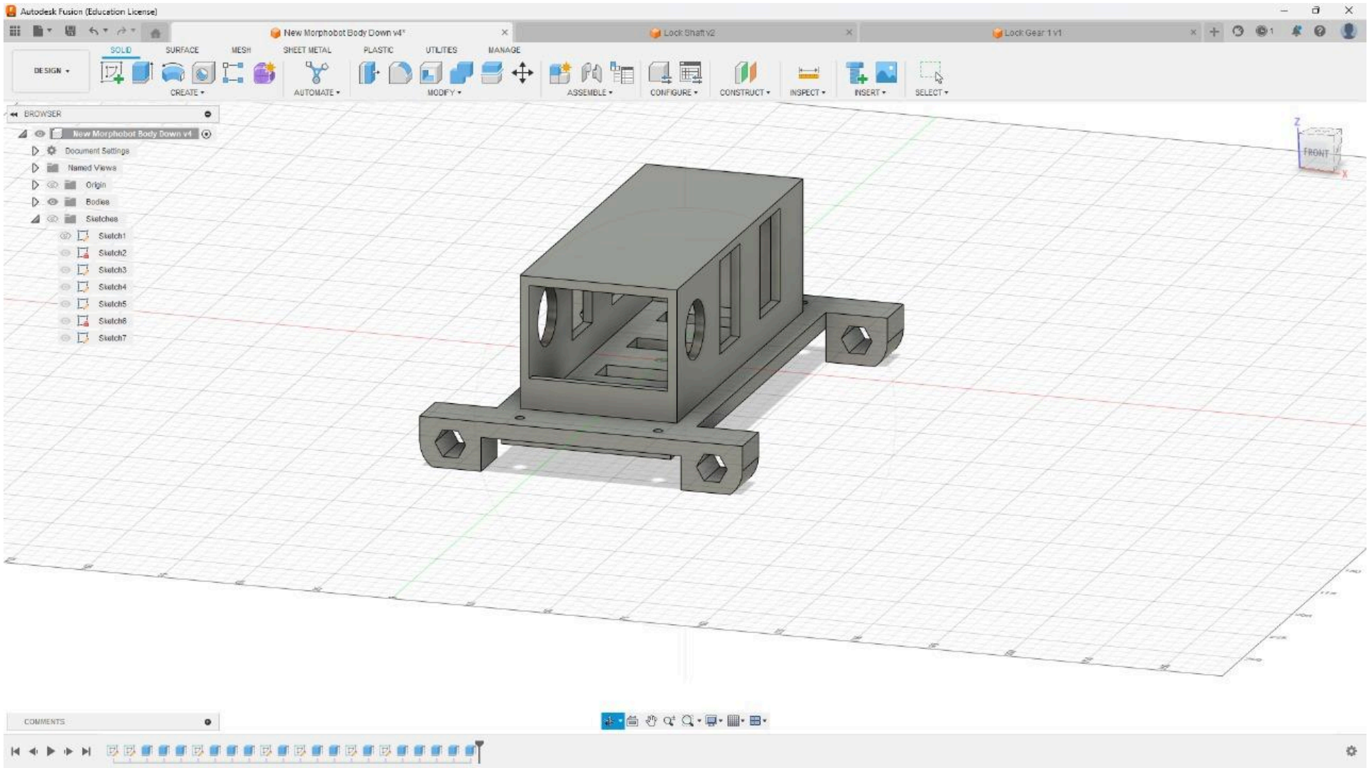


Fig. 5.1

The body of Morphobot is the main structure that holds the entire robot together. It's designed to be strong and stable, which is important for both moving on the ground and flying. The body has special spots for attaching things like motors and control systems, making it easy to put together and fix if needed. Its design helps Morphobot move smoothly and handle different tasks, whether it's traveling over rough terrain or flying through the air. This makes the robot versatile and able to adapt to many situations.

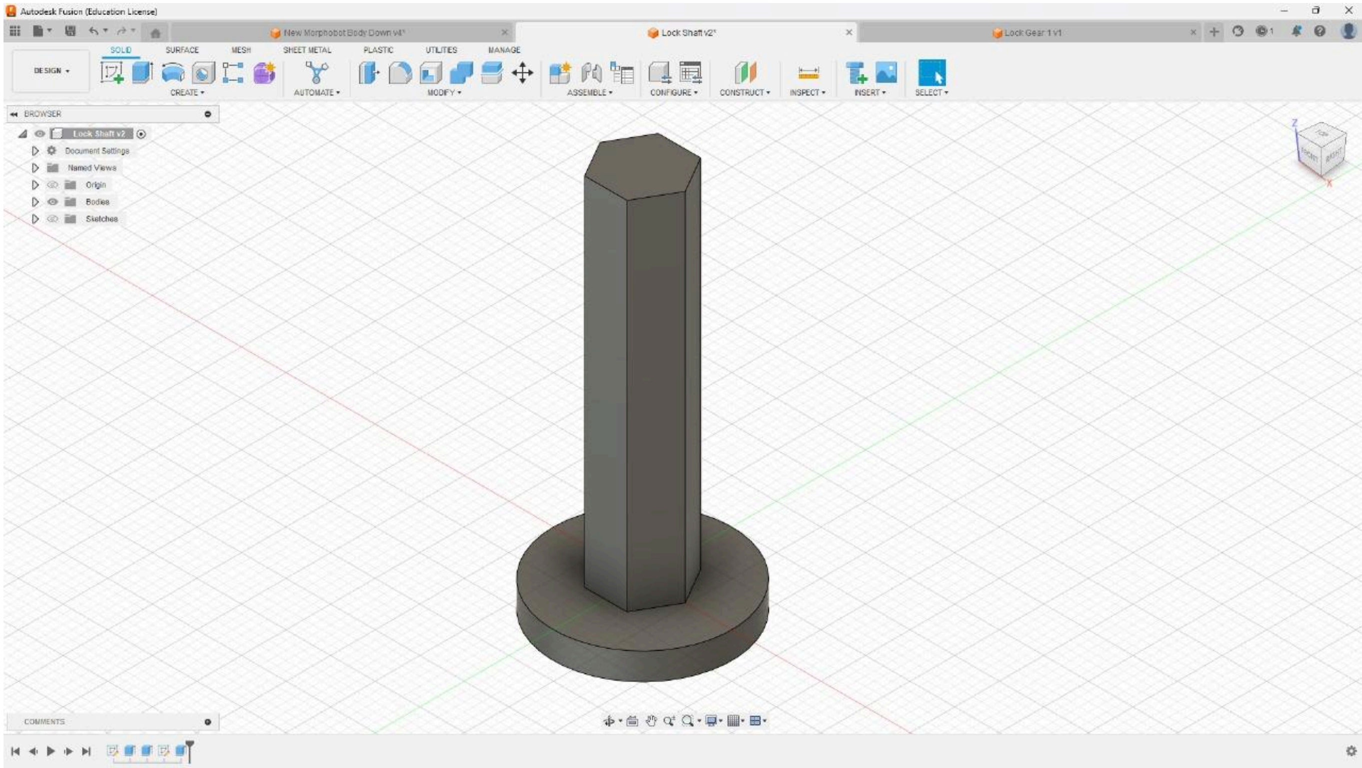


Fig. 5.2

This image shows a hinge part used in Morphobot. This hinge locks the arm with the main body and helps the Morphobot move parts of its body smoothly, which is especially important when it changes from driving to flying mode. It allows the parts to turn or pivot safely, making sure the robot can switch its setup quickly and smoothly. The hinge is also strong, helping Morphobot handle different tasks and environments without breaking.

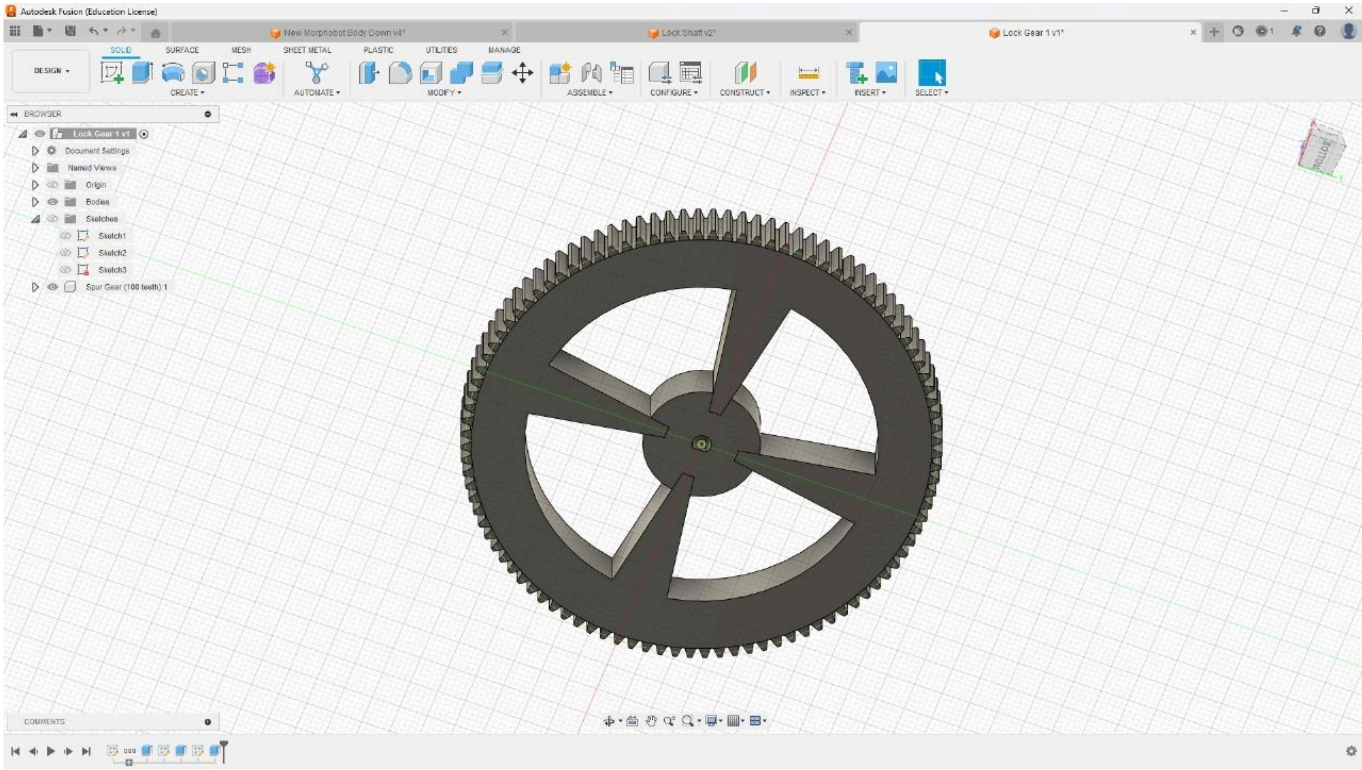


Fig. 5.3

This image shows a gear wheel used in Morphobot. The gear wheel is important because it helps Morphobot move parts of its body or change how it moves. This particular design ensures that Morphobot can perform its tasks smoothly, whether driving or flying, by helping to control the speed and direction of its movements.

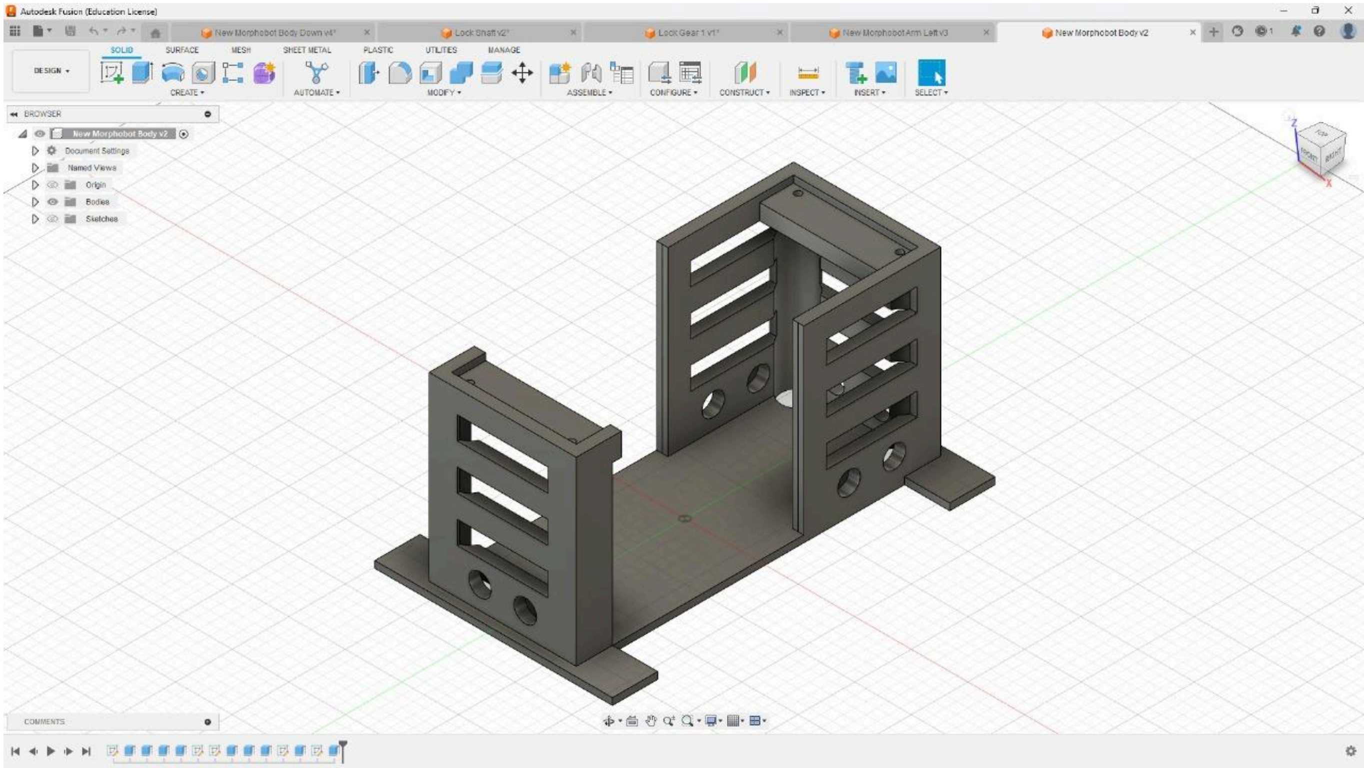


Fig. 5.4

This image shows a part of Morphobot's body, specifically a frame structure. This frame is designed to support and protect the internal components of the robot, like its electronics and motors. It also has multiple slots and mounting points, making it easy to attach different parts or modules as needed. This flexibility is important because it allows Morphobot to be customized for various tasks. The robust design of the frame ensures that it can handle different environments, whether Morphobot is moving on the ground or flying, keeping everything inside safe and secure.

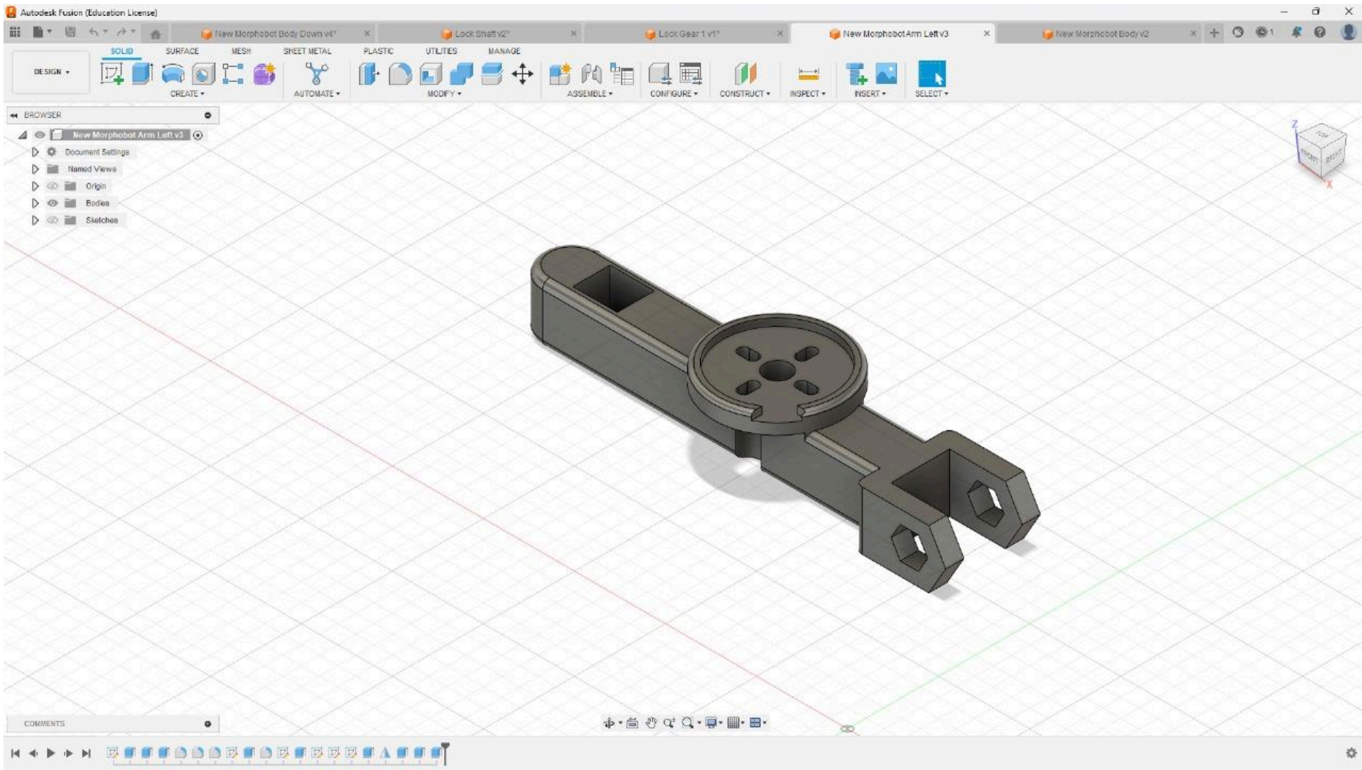


Fig. 5.5

This image shows a support arm of Morphobot, which is essential for providing stability and support for the wheels. This arm is specifically designed to hold the wheels in place, ensuring that Morphobot can move smoothly and maintain balance, especially when transitioning between different terrains or modes of operation. Its sturdy construction helps to bear the weight and stress of movement, contributing to the overall durability and functionality of Morphobot.

6. Code

```
import RPi.GPIO as GPIO
import serial
import time
from time import sleep

in1_motor1 = 24
in2_motor1 = 23
en_motor1 = 25

in1_motor2 = 17
in2_motor2 = 27
en_motor2 = 22

GPIO.setmode(GPIO.BCM)
GPIO.setup(in1_motor1, GPIO.OUT)
GPIO.setup(in2_motor1, GPIO.OUT)
GPIO.setup(en_motor1, GPIO.OUT)
GPIO.output(in1_motor1, GPIO.LOW)
GPIO.output(in2_motor1, GPIO.LOW)
p_motor1 = GPIO.PWM(en_motor1, 1000)

GPIO.setup(in1_motor2, GPIO.OUT)
GPIO.setup(in2_motor2, GPIO.OUT)
GPIO.setup(en_motor2, GPIO.OUT)
GPIO.output(in1_motor2, GPIO.LOW)
GPIO.output(in2_motor2, GPIO.LOW)
p_motor2 = GPIO.PWM(en_motor2, 1000)

p_motor1.start(25)
```

```
p_motor2.start(25)

bluetooth_port = '/dev/ttyAMA0'
baud_rate = 9600

bluetooth = serial.Serial(bluetooth_port, baud_rate)
time.sleep(2)

print("R-right L-left S-stop F-forward B-backward W-low U-medium V-high
e-exit\n")

while True:
    x = bluetooth.read().decode('latin-1')

    if x == 'F':
        print("forward")
        GPIO.output(in1_motor1, GPIO.LOW)
        GPIO.output(in2_motor1, GPIO.HIGH)
        GPIO.output(in1_motor2, GPIO.LOW)
        GPIO.output(in2_motor2, GPIO.HIGH)
        x = 'z'

    elif x == 'B':
        print("backward")
        GPIO.output(in1_motor1, GPIO.HIGH)
        GPIO.output(in2_motor1, GPIO.LOW)
        GPIO.output(in1_motor2, GPIO.HIGH)
        GPIO.output(in2_motor2, GPIO.LOW)
        x = 'z'

    elif x == 'R':
        print("right")
        GPIO.output(in1_motor1, GPIO.LOW)
        GPIO.output(in2_motor1, GPIO.HIGH)
```

```
        GPIO.output(in1_motor2, GPIO.HIGH)
        GPIO.output(in2_motor2, GPIO.LOW)
        x = 'z'

    elif x == 'L':
        print("left")
        GPIO.output(in1_motor1, GPIO.HIGH)
        GPIO.output(in2_motor1, GPIO.LOW)
        GPIO.output(in1_motor2, GPIO.LOW)
        GPIO.output(in2_motor2, GPIO.HIGH)
        x = 'z'

    elif x == 'S':
        print("stop")
        GPIO.output(in1_motor1, GPIO.LOW)
        GPIO.output(in2_motor1, GPIO.LOW)
        GPIO.output(in1_motor2, GPIO.LOW)
        GPIO.output(in2_motor2, GPIO.LOW)
        x = 'z'

    elif x == 'W':
        print("low")
        p_motor1.ChangeDutyCycle(25)
        p_motor2.ChangeDutyCycle(25)
        x = 'z'

    elif x == 'U':
        print("medium")
        p_motor1.ChangeDutyCycle(50)
        p_motor2.ChangeDutyCycle(50)
        x = 'z'

    elif x == 'V':
        print("high")
```

```
    p_motor1.ChangeDutyCycle(75)
    p_motor2.ChangeDutyCycle(75)
    x = 'z'

elif x == 'e':
    GPIO.cleanup()
    print("GPIO Clean up")
    break

else:
    print("<<<  wrong data  >>>")
    print("please enter the defined data to continue.....")
```

- **Bluetooth-Based Motor Control for Morphobot:** This Python code controls a robot's motors via Bluetooth commands, allowing it to move forward, backward, turn left or right, and stop. It also lets you adjust the speed of the motors to low, medium, or high levels. The purpose of this code is to provide easy and flexible control over the robot's movement, making it highly adaptable for various tasks and environments. This setup is particularly useful for remote operations where direct physical control isn't feasible.

7. Results and Discussion

Throughout the testing phase, Morphobot demonstrated a high level of adaptability and efficiency in various environments, successfully navigating complex terrains. Its dual-mode functionality allowed seamless transitions between ground and aerial movements, showcasing its capability to handle diverse operational demands.

Performance in Different Terrains: On the ground, Morphobot moved smoothly, with its specially designed wheels providing excellent traction on both smooth surfaces and rough outdoor terrains. The wheel's durability was evident, as it withstood extensive use without significant wear, maintaining optimal performance throughout the tests.

Aerial Maneuverability: In flight mode, Morphobot exhibited stable and controlled movements. The transition from land to air was smooth, with the robot quickly adapting to aerial navigation. This capability proved particularly useful in scenarios where ground navigation was hindered by highlighting Morphobot's versatility.

Challenges Faced: Despite its successes, there were challenges, primarily related to power management. The demands of switching between two modes required considerable energy, which sometimes led to shorter operational periods than anticipated. However, adjustments to power allocation strategies saw improvements in subsequent tests.

8. Conclusion

In conclusion, the Morphobot project has successfully demonstrated a highly adaptable robot that can seamlessly transition between land and air travel. This capability allows it to effectively navigate diverse environments, from urban landscapes to natural terrains. The integration of a Raspberry Pi 5 for ground control and an Arduino for aerial maneuvers has proven effective, offering both manual and autonomous operation modes.

Morphobot has proven that it can adapt well to various environments, making it a great tool for collecting data and exploring areas that are tough to reach. Its ability to switch between driving and flying easily is especially helpful.

MORPHOBOT PHASE-1

MORPHOBOT PHASE-2

MORPHOBOT PHASE -3