EECS/BioE/MechE 106A/206A

D.A.R.T: Driving And Rising Technology

Fall 2024

1 Team Information

Name	Background
Smit	Smit is a 4th year EECS major interested in Power Electronics, EV and Energy systems, electronics hardware design, and some firmware.
Amal	Amal is a 4th year EECS major interested in electronics and hardware design, pcb design, and robotics.
Nikhil	Nikhil is a 4th year EECS major interested in low-level programming, computerized systems infrastructure, and HV system design.
Andrew	Andrew is a 3rd Year Aerospace Engineering and EECS double major, interested in robotics, mechanical and electrical HW.
Mary	Mary is a 4th year EECS and Bioengineering double major interested in computer architecture and hardware design.

2 Abstract

Our project aims to develop a transformable robotic vehicle that can transform from a car to a drone. It will achieve this by rotating the wheels 90 degrees and take flight using built-in propellers. It will be equipped with ultrasonic sensors to detect obstacles in its path, and navigate around them, either by maneuvering on the ground or flying over it. This innovative design offers the potential for enhanced mobility and versatility in various applications, such as search and rescue operations, surveillance, and package delivery.

3 Project Description

3.1 Project Goals

What are your project goals?

- Successful Transformation: Develop a mechanism that allows the vehicle to reliably transform between car and drone configurations.
- Obstacle Detection and Avoidance: Implement an effective obstacle detection system using ultrasonic sensors to enable the vehicle to navigate around obstacles both on the ground and in the air through sensing and planning.
- Optimization: utilize weight-saving methods and implement power optimization to ensure the robot is as efficient as possible to maximize operation time.

3.2 Project Architecture

DART's Hardware and Software systems will work together to meet project goals specified in section 3.1.

3.2.1 Software

- ROS: Using ROS Client/Server or Subscriber Publisher, to communicate sensor information to the micro controller to make high-level decisions for e.g if DART should be in drive mode or fly mode. And for the motor controllers to make more fine decisions like controlling power to each motor.
- Communication Protocols: We have the architecture for ROS setup as mentioned above but we will use some protocols to communicate this information on the firmware level. Some protocols we may use include:
  - SPI
  - I2C
  - UART
  - Analog and PWM
- Object detection libraries - to use information from the camera and Distance sensors to map objects and make decisions to avoid them.
- Autonomous and Remote Operation Modes
  - Remote Operation: Radio-controlled mode will allow an operator to manually control the robot's movement and transformation.
  - Pre-programmed Path Execution: Users can define paths using way-points; the software will switch modes (drive/fly) as needed along the route
- Control Loops: PID Controllers will manage the speed of motors for smooth driving and flight stability

3.2.2 Hardware

The system will consist of the following subsystems

- Computing Subsystem: Responsible for coordinating and controlling other subsystems.
  - Remote Operation: executes commands transmitted over the communication subsystem, interrupting them when necessary based on sensor subsystem feedback.
  - Autonomous Operation: executes pre-programmed commands while reacting to feedback from the sensor subsystem.
- Power Distribution Subsystem: Responsible for ensuring sufficient power is provided to all subsystems when needed.
- Sensor Subsystem: Responsible for providing information about the robot's local environment; detecting obstacles, mapping surroundings, etc. Consists of the following sensors:
  - Digital Camera
  - Distance Sensors
  - IMU (Inertial Measurement Unit)
- Communication Subsystem:
  - Remote Operation: Receives command inputs, transmits sensor information and other subsystem data.
  - Autonomous Operation: Transmits sensor information, and other subsystem data.
- Flying Subsystem: Consists of a set of 4 motors in a quad-copter configuration. Works with the computing subsystem and sensor subsystem to fly in the air.
- Driving Subsystem: Consists of a set of 4 motors to drive on the ground.
- Transformation Subsystem: Responsible for transitioning between the flying and driving configurations of the robot.

3.3 Project Sensing, Planning, and Actuation

How does your project incorporate sensing, planning, and actuation?
Sensing:

- **Drive Mode**: Sensors like LiDAR or ultrasonic sensors will be used to detect obstacles, focusing on the vertical axis in front of the robot. These sensors will determine the height of obstacles and detect when they become unavoidable.
- **Flight Mode**: The same sensors will monitor the height and proximity of obstacles, determining how high the drone needs to fly to clear them. Additionally, these sensors will identify when the obstacle has been cleared, allowing the robot to safely return to the ground. Sensors will monitor the terrain below during flight to detect when it is safe to land (e.g., flat surface or low obstacle height).

Planning:

- **Drive Mode**: The robot will follow a pre-specified path or respond to remote control inputs for direction. While navigating, the robot will analyze the sensor data to detect obstacles in its path. If an obstacle is too tall to drive over or around, the robot's planning system will trigger a transition to flight mode.
- **Flight Navigation**: The planning system will use obstacle height data to calculate the necessary altitude for the drone to fly over the obstacle safely. The system will continuously monitor the vertical space until it senses that the obstacle has been passed and there's a clear landing zone. Once the planning system detects that the obstacle is behind the robot and that it can safely descend, it will initiate the transition back to drive navigation, allowing the robot to resume following the path or remote control directions.

Actuation:

- **Drive Mode**: The robot's drive motors will actuate based on planned paths or remote control inputs. When the system senses an unavoidable obstacle, the robot will stop moving, preparing for transition into flight mode.
- **Flight Mode**: The robot will actuate propeller motors to take off and adjust its altitude based on the calculated obstacle height. The actuation system will maintain flight until the obstacle is passed, at which point the descent will be triggered. The actuation system will control the drone's descent once the terrain below is deemed safe for landing, switching back to drive mode.

3.4 Project Tests and Metrics

How will you test or assess your project? What constitutes a success? What are some realistic goals? What are some reach goals?
This project is intended as mainly a proof-of-concept. That is, a successful project will involve basic testing of the robot's core functions. Enumerated, these are:

- Motor control: Drive all motors simultaneously at varying speeds using the MCU. This validates the motors are functional, the MCU can send commands, and the power delivery system to the motors is adequately sized.
- Communication: Drive one or more motors over the radio control system, testing the radio. Repeat this test for multiple ranges to find a maximum radio communication distance.
- Basic driving: The robot drives a short distance without flying, from commands issued through the radio controller.
- Basic flying: The robot flies a short distance (take off, hover, rotate, directional translation, and landing) without driving from commands issued through the radio controller.

- Drive → Fly transition: Start by driving, transition to flying, and end by flying.
- Fly → Drive transition: Start by flying, transition to driving, and end by driving.
- Pre-programmed path execution: Send a path to the vehicle from the base station with instructions to drive/fly a certain distance, then execute those instructions with reasonable accuracy.
- Driving with sensing: Drive with automatic object avoidance
- Flying with sensing: Fly with automatic object avoidance

These are all reasonable goals which should be able to be completed within the given time window. A stretch goal would be to implement an automatic fly/drive transition feature, where the vehicle could transform from flying to driving mode (or vice versa) based on the environment in order to avoid obstacles. A test for this may be a set of stairs, in which the vehicle would detect the stairs and transition to flying mode to avoid said stairs by going over them.

4 Tasks

Here, list out different major and minor tasks of the project, along with when you plan to achieve them. Keep in mind the checkpoint dates (11/9 and 11/29). For example,

1. **Build the robot**
   (a) **Design the Robot** [by 11/2]
      i. **Design the Robot Frame** [by 11/2]
         A. **Design the Transformation Subsystem** Emphasis is put on ensuring a sound transformation system is designed, other parts of the frame are dependent this design.[by 11/1]
         B. **Design Other Robot Structural Components** Ensure the system is structurally sound in both modes of operation, plan out the placement of electronics. [by 11/1]
      ii. **Design the Robot Electronics** [by 11/2]
         A. **Select Necessary Parts** Spec out the exact components used for the robot, crucial to determining other aspects of design. [by 10/26]
         B. **Design the Central PCB** Based on component specifications, incorporate Power Distribution, Computing, Communication, and Sensor subsystems onto a unified PCB. [by 11/2]
   (b) **Assemble the Robot** [by 11/11]
      i. If any issues are encountered in this stage, rapid prototyping will modify the design.
2. **Code the robot.** We will program the robot using C++, Python, Arduino
   (a) **Develop firmware for peripherals listed below** [by 11/6]
      i. IMU
      ii. Motor controller
      iii. Servo controller
      iv. Camera and Distance sensors
   (b) **Develop ROS Nodes** : Controller and sensor nodes to broadcast information [by 11/13]
   (c) **Develop Software/firmware** for controlling DART in Remote mode [by 11/25]
   (d) **Object detection and path planning algorithm**: Working in parallel to Firmware development to design the autonomous planning. [by 11/25]
3. **Testing**

(a) **Hardware Validation** Ensure all the hardware is working as expected, without direct software control from intended controller system [by 11/17]

(b) **Software Validation** Ensure software is functioning without intended hardware system (HIL testing or similar) [by 11/25]

(c) **Integration Testing** Integrate software and hardware together, and perform tests as described in the "Testing" section [by 12/2]

(d) **Stretch Goals** Implement anything additional and perform tests for those features [by 12/3]

5 Bill of Materials

5.1 Use of Lab Resources

This project will largely be constructed without lab resources. However, if possible we would like to request a LiDAR and/or lightweight near field sensing camera of some kind if possible.

Item	**Quantity**
LiDAR / lightweight camera	1

5.2 Other Robotic Platforms

We do not have access to any other external robotic platforms (nobody in our group is a member of a robotics-related lab).

5.3 Other Purchases

The majority of components for this project will need to be purchased. We expect this will exceed the $60 budget, but understand that we will be responsible for covering the remainder of the purchases. The total estimated project budget is ≈$100.
Components are listed as general categories to best communicate the types of items required. Note that specific part numbers will be collated for submission in the ESG form (up to the $60 amount).

Item	**Quantity**	**Justification**
Fly Motors	4	Brushless DC (BLDC) motors to drive the propellers to create flight
Drive Motors	4	BLDC motors to drive the wheels to enable motion on the ground
Battery	1	LiFePo4 sized to power fly/drive motors, sensors, servos, and MCU
Radio Transceivers	2	Communicate between vehicle and base station (controller)
Servos	4	Adjust wheels from flying $\iff$ driving
Wiring	N/A	Harnessing for tying everything together
PCB	1	Mainboard for logic controllers to simplify harnessing
Microcontroller (MCU)	1	Main logic controller (Ardunio/Teensy)
Inertial Measurement Unit (IMU)	1	Orientation and gyroscope for flight
Motor Controllers	8	Drives motors with output from MCU
Ultrasonic Sensors	4-8	Distance sensing for each side; octagonal approximation
FPV Camera (optional)	1	If weight allows, add for human sight
Frame Materials	N/A	3D printed frame out of PETG or PLA
Wheels	4	Enables motion; 3D printed wheels out of TPU
Propeller	4	Enables lift

6 Other

This section contains all additional information necessary to convince us that *a)* you are equipped to complete the project you propose, and *b)* you have thought specifically about your project implementation. This section is optional, but some things you might mention include:

- ROS packages you'll need (with pointers to relevant websites);
- preliminary code structure/skeleton; and
- mechanical designs/drawings/sketches of your project.

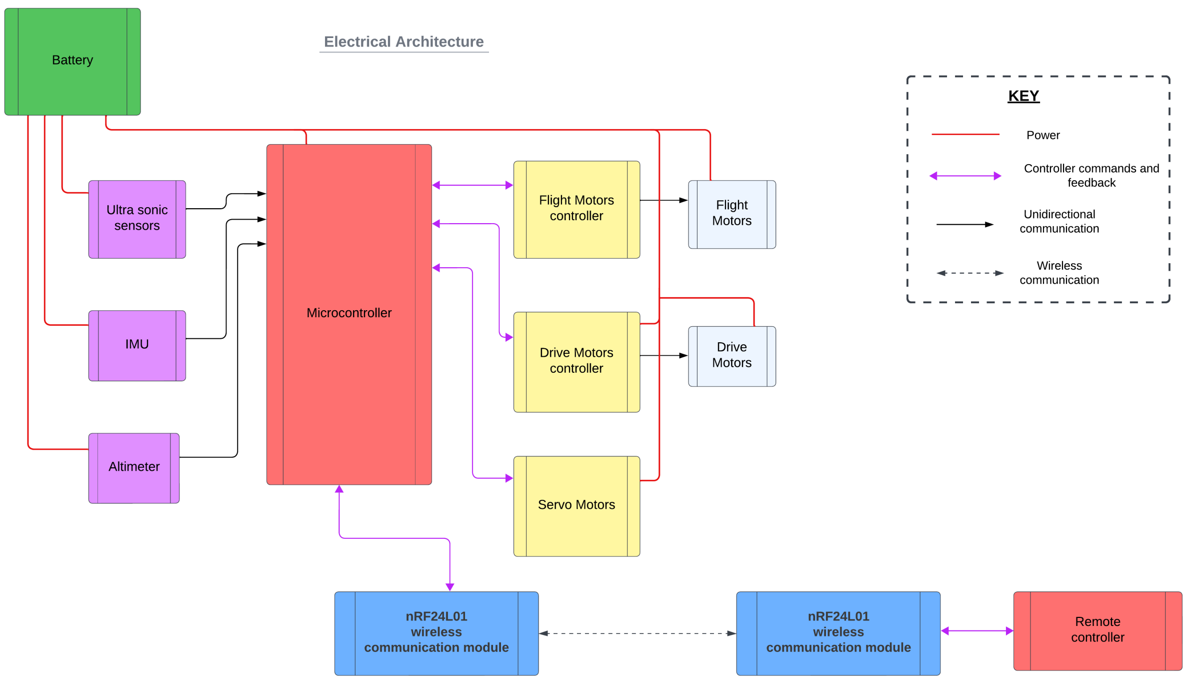

Figure 1: Electrical Architecture

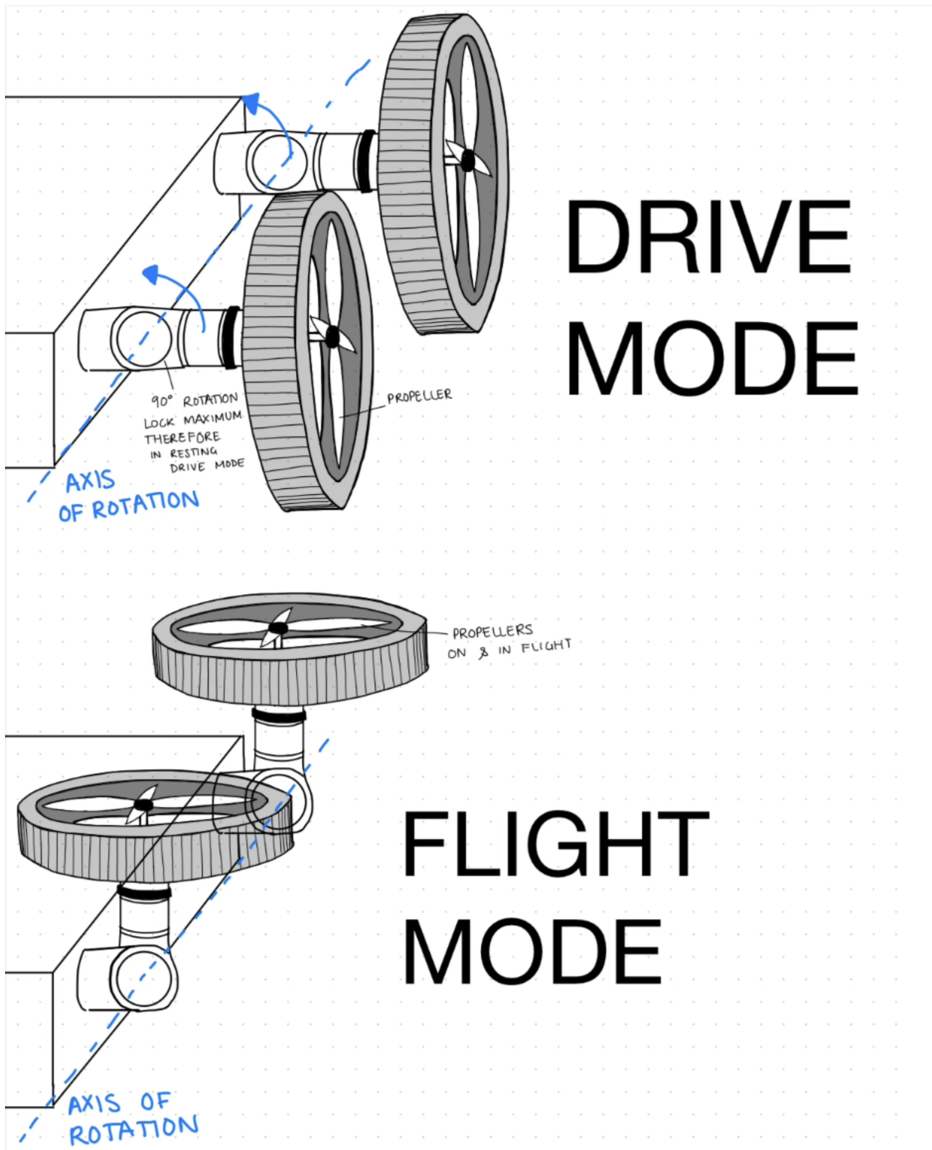

Figure 2: Drive Mode vs Flight Mode Joint Positions