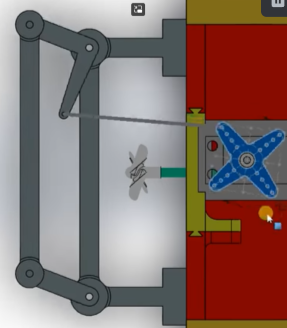
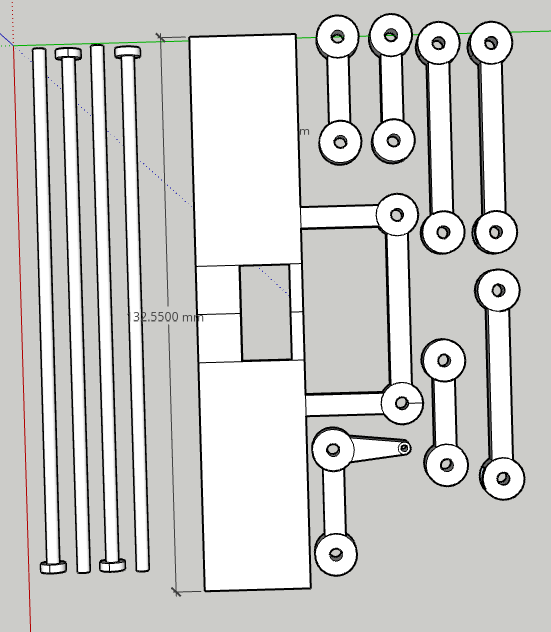
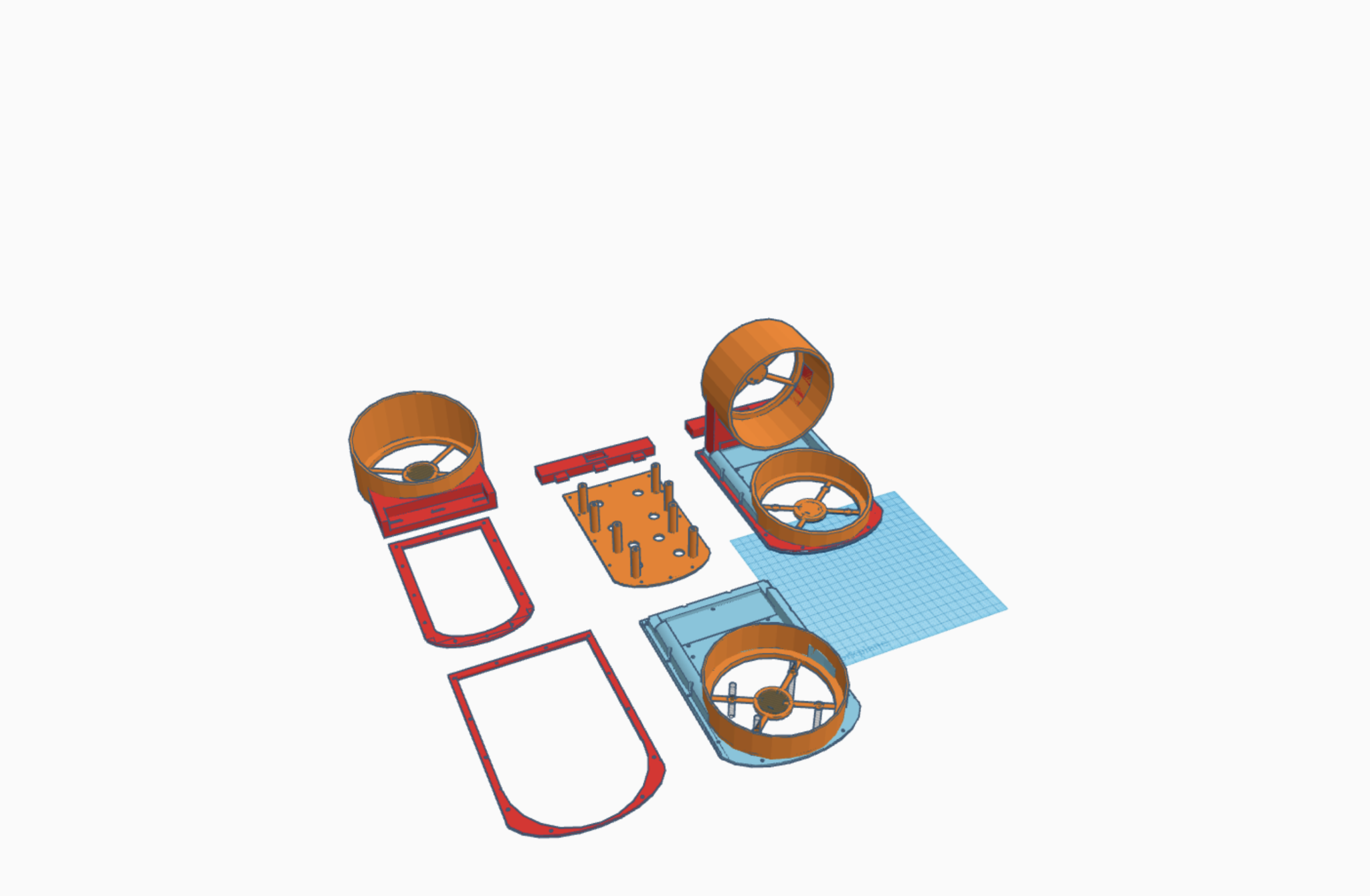
WalkAlong Hovercraft Report

**Product Description**

The WalkAlong Hovercraft is a functional teaching tool to teach new students about airflow emphasizing the teachings of Bernoulli’s principle through its lipped propellor ducts, air chamber propulsion, and its rudder control. A hovercraft is a vehicle that operates using a cushion of air, we have decided to downsize a larger design to make a more controllable, responsive, and articulate model to play with the air cushion concept. A hovercraft works by using a skirt that creates a slightly higher pressure chamber of air below compared to the atmospheric air around it, creating lift and allowing it to float above the surface. The back fan uses air as thrust to push forward following newtons 3rd law, pushing back to move the hovercraft forward. The back fan also presents an opportunity to teach about adverse yaw in fluid mediums such as air.

**Design**



Our hovercraft uses a heavy-duty controller and power source of two twelve-volt battery packs to power two DC motors which are situated below and in the back as the main propulsion and thrust tool. We chose to emphasize the air cushion concept by running a long tow line connecting the heavy power supply which allows the user to drag the hovercraft along and experience weightlessness. To steer the hovercraft, we used a servo-controlled rudder to allow the user to experiment with airflow. The rudder design is based on the rightmost diagram with a servo and a string controlling the yaw of the rudder.

**3D Print/Design Files**

Base Links:

[3D Design Two Fan Hovercraft Base | Tinkercad](https://www.tinkercad.com/things/kVbcMpU3Tm1-two-fan-hovercraft-base/edit?sharecode=U488YAltxMp5n4AVxRoQ5BAR0rwmgUgtBMamYxmB1GM)

Rudder Links (Rudder Parts and Rudder):

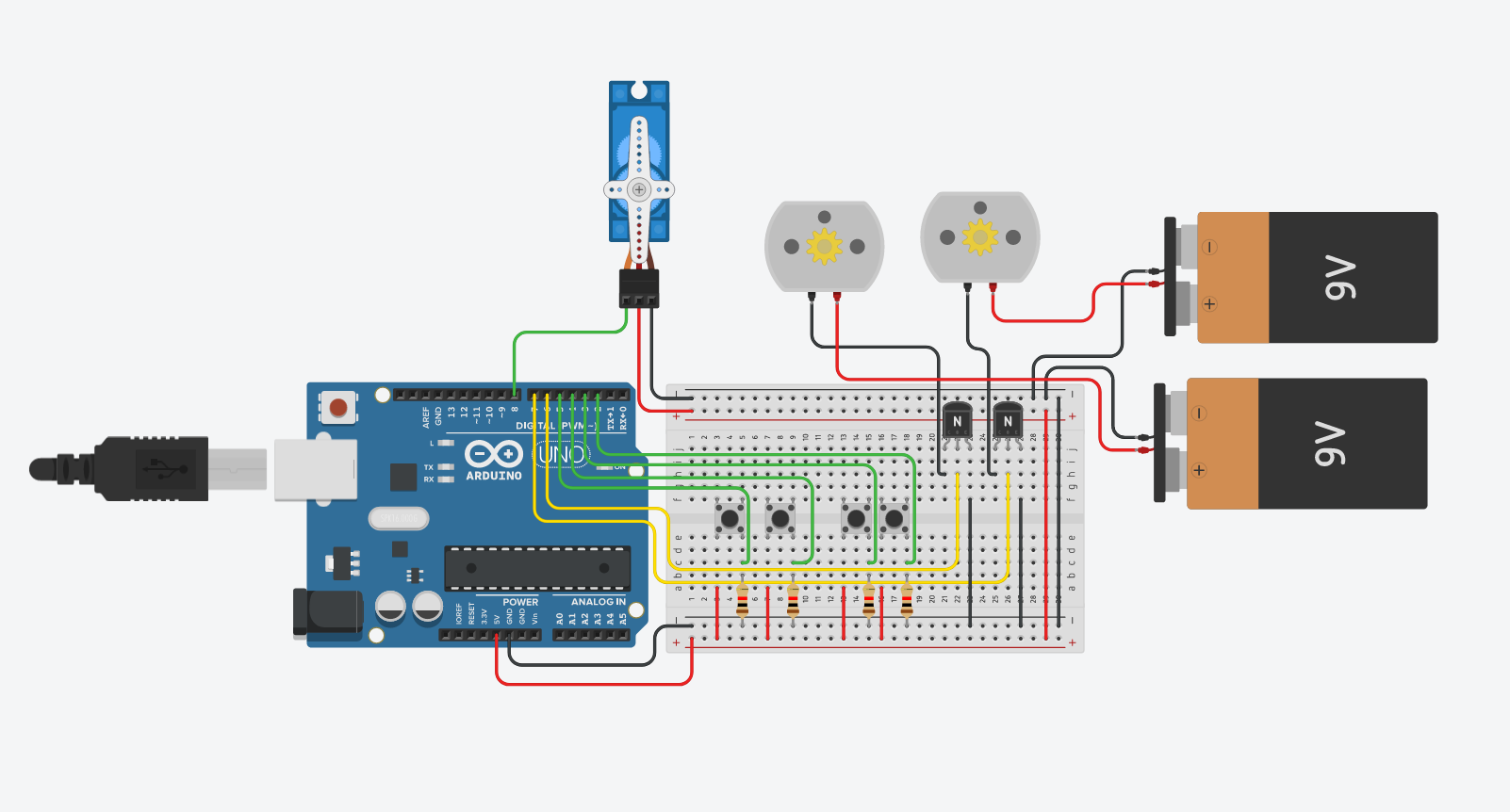
<https://app.sketchup.com/share/tc/northAmerica/MzMMjym1v-M?stoken=JnlcC-zclRR9P7GmzeXpG4HWIfSawc0zNsIAbrqyPCjS62iiZE_-m0ie5Y0iT0cy&source=web>

<https://app.sketchup.com/share/tc/northAmerica/OIbSJxfIqeE?stoken=wMK6SELZr9p6nOJLHTnUqf50F7EqTHqCCenG_asq3cvhbJW-3VqutmqvlklLTdVX&source=web>

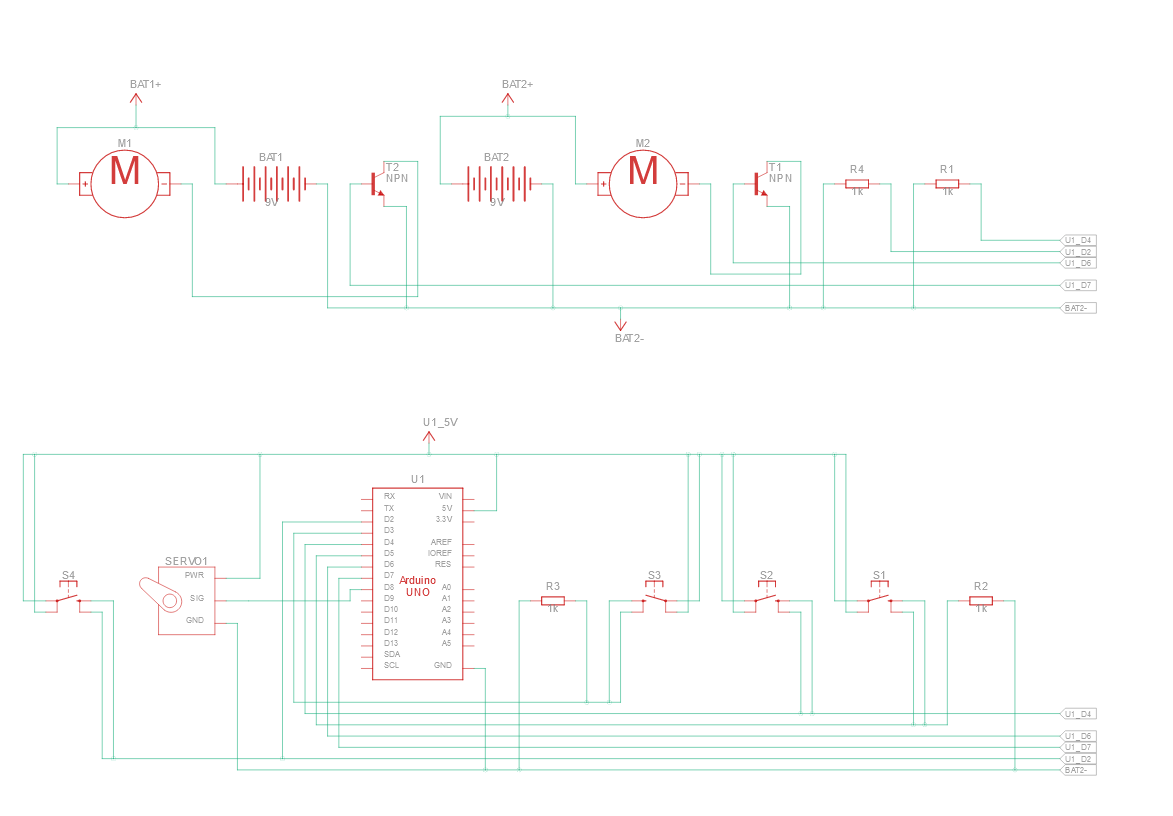
**Parts**

* **Skirt**
  + Light Waste Disposal Bag **x1**
* **Fans**
  + 12 cm Diameter Fans **x2**
  + DC Hobby Motor **x2**
  + Transistor **x2**
  + Push Button **x2**
* **Steering**
  + Micro Servo Motor **x1**
  + Arduino Uno MicroController **x1**
  + Push Button **x2**
  + Breadboard **x1**
* **Power**
  + 12 Volt AA Battery Packs **x2**
* **3D Prints**
  + 3D printed Fan Duct **x2**
  + Base Plate **x1**
  + Base Skirt Clamp **x1**
  + Skirt Bottom Outline **x1**
  + 4.5 cm Tall Rudder **x2**
  + Supporting Rudder Parts and Housing **x7**
* **Assembly** 
  + Wooden Skewers for Rudder Support **x4**
  + 1cm Screws **x12**
  + 8 Flat Head 1cm Screws **x8**

**Wiring Diagram:**



**Circuit Schematic:**



Legend:

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Name | Component | Name |
|  | DC Motor |  | Push Button |
|  | Arduino Uno |  | Servo Motor |
|  | Battery |  | Resistor |
|  | Transistor |  |  |

**Prototype:**

Video Demo of Our Prototype:

[**https://www.youtube.com/watch?v=3dcQ3JC8JqY**](https://www.youtube.com/watch?v=3dcQ3JC8JqY)

**TroubleShoot Log:**

|  |  |  |
| --- | --- | --- |
| **Issues/Problem** | **Attempted Solutions** | **Final Solution** |
| - Computer Fan does not have enough rpm | - Switch the fan from a computer fan to a DC motor that we found | - New DC fan increased the overall rpm  - causes more Air to be produced by the fan to make it hover better |
| - Not having enough space to fit the whole height of the DC motor and fan into the hovercraft duct  - this caused a lot of air to blow out instead of down | - We initially started with a design where we made the ducts taller but this would have caused a lot of extra weight | - we created a hole in the middle of the fan duct holder so we could slide the fan lower into the hovercrafts base  - this caused the fan to be inside the duct and caused all the air to be blown down and not get blown out to the sides |
| - You can’t turn a DC motor on and off alone using an arduino uno when using a 12 V power source | - we tried adding transistors to be able to transfer the motors power from the 12V and turn it on and off when a button is pressed | - this allowed us to have an easy way to turn the motor on and off using the arduino through buttons |
| - The back of the hovercraft was too heavy and was leaning to the back to much | - first we removed the big and heavy duct from the back of the craft and replaced it with a lighter holder for the back fan | - removed duct and reduced the skirt size so there was less space for the hovercraft to tilt  - this caused the back side to tilt over less and with the less weight on the back the hovercraft could move much easier without this resistance holding the back down |
| - One fan was not strong enough to lift and move the hovercraft | - at first we tried to move the hovercraft with on fan blowing air below the craft and to the back of the hovercraft | - We then changed this so that we have one fan blowing down that blows up the hovercraft skirt and lets it hover and another fan the will move the hovercraft forwards  - With this new fan design we were eventually able the make the hovercraft float and move after applying 12 volts to both the fans  - it's fair to say that with one fan the hovercraft would have not been able to float |

**Arduino Uno Code:**

#include <Servo.h>

Servo myservo; //creates servo object to control a servo

int dcButtonOne = 5;

int dcButtonTwo = 4;

int servoButtonLeft = 3;

int servoButtonRight = 2;

int dcMotorOne = 6;

int dcMotorTwo = 7;

int servoPosition = 90; //the servos starting position

void setup() {

pinMode(dcMotorOne, OUTPUT);

pinMode(dcMotorTwo, OUTPUT);

pinMode(dcButtonOne, INPUT);

pinMode(dcButtonTwo, INPUT);

pinMode(servoButtonLeft, INPUT);

pinMode(servoButtonRight, INPUT);

myservo.attach(8); //attaches the servo to pin 9

Serial.begin(9600);

}

void loop() {

//If this button is pressed, DC motor one will be turned on by the transistor

if(digitalRead(dcButtonOne) == HIGH){

analogWrite(dcMotorOne, 200);

Serial.println("MotorOne On");

//If this button is not pressed, DC motor one will be turned off by the transistor

} else {

analogWrite(dcMotorOne, 0);

Serial.println("MotorOne Off");

}

//If this button is pressed, DC motor two will be turned on by the transistor

if(digitalRead(dcButtonTwo) == HIGH){

analogWrite(dcMotorTwo, 200);

Serial.println("MotorTwo On");

//If this button is not pressed, DC motor two will be turned off by the transistor

} else {

analogWrite(dcMotorTwo, 0);

Serial.println("MotorTwo Off");

}

if(digitalRead(servoButtonLeft) == HIGH && servoPosition + 1 != 181){ //checks if the left button has been press and the servo can move more to the left

servoPosition += 1;

myservo.write(servoPosition); //moves the servo to the servoPosition int

delay(15); //waits for the servo to move

Serial.println("Servo Moving Left");

} else if(digitalRead(servoButtonRight) == HIGH && servoPosition - 1 != -1)

servoPosition -= 1;

myservo.write(servoPosition); //moves the servo to the servoPosition int

delay(15); //waits for the servo to move

Serial.println("Servo Moving Right");

}