DON'T CRASH!

Autonomous Motor Response to Getting too Close

Nicolas Lai

Boston University

Physical Electronics Lab Open House

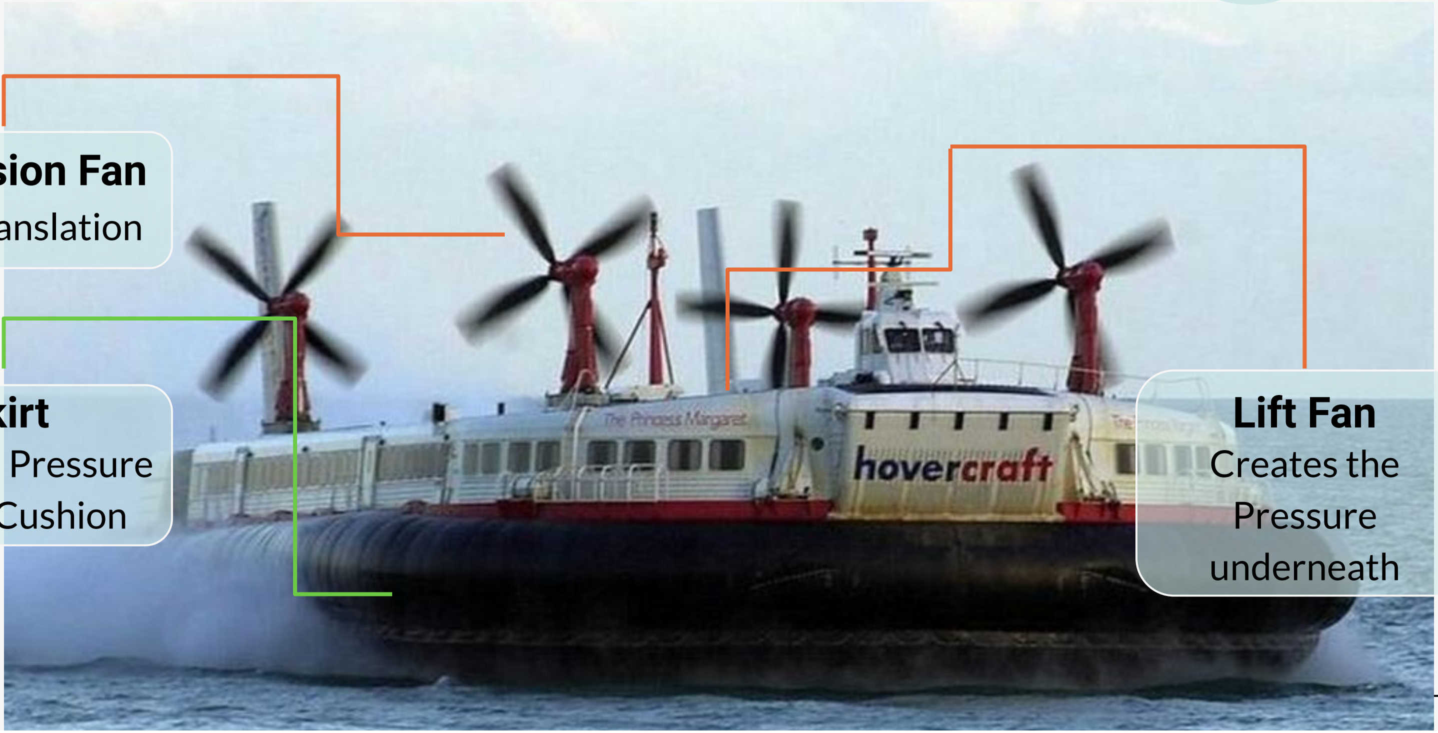
April 29th, 2021

OBJECTIVE

Don’t Crash!!!

Prevent collision with motor shutdown.

HOVERCRAFT OVERVIEW

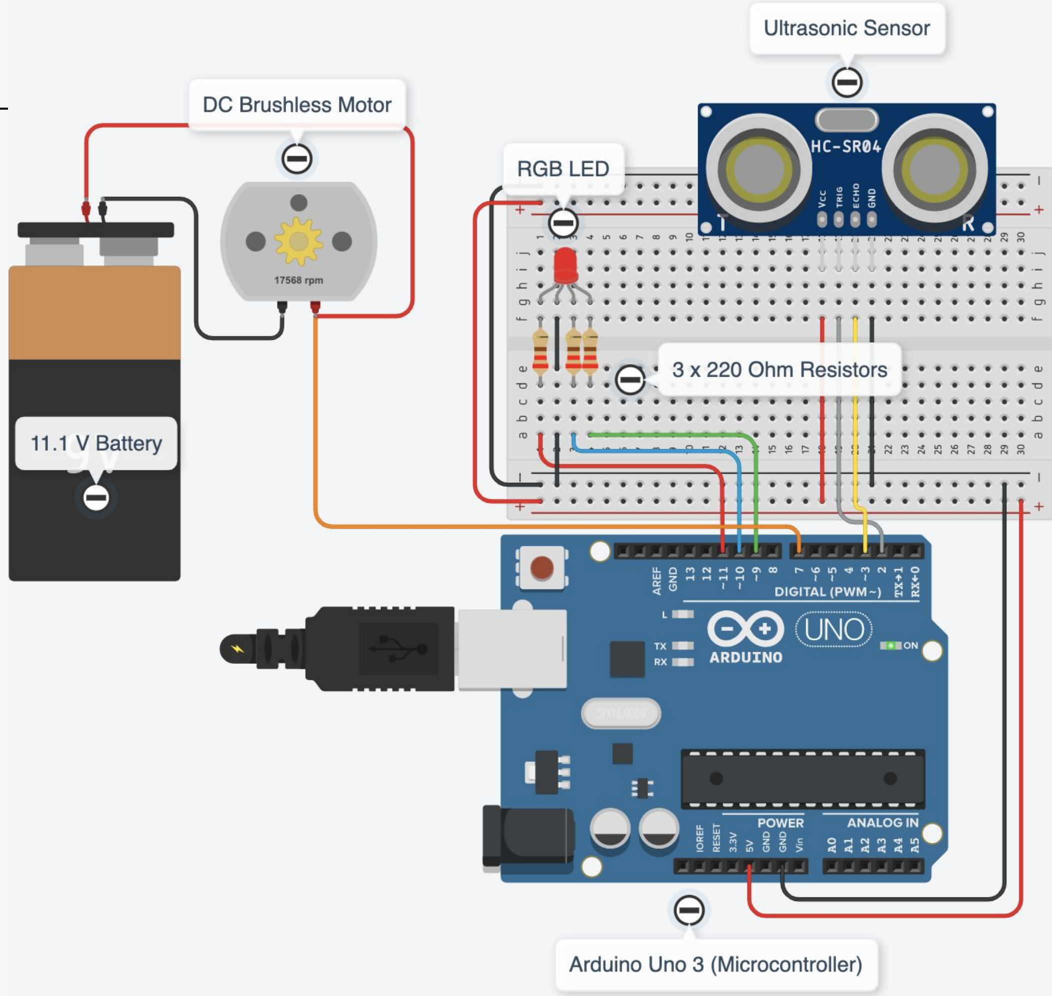


Propulsion Fan
X & Y Translation

Skirt
Maintain Pressure for Air Cushion

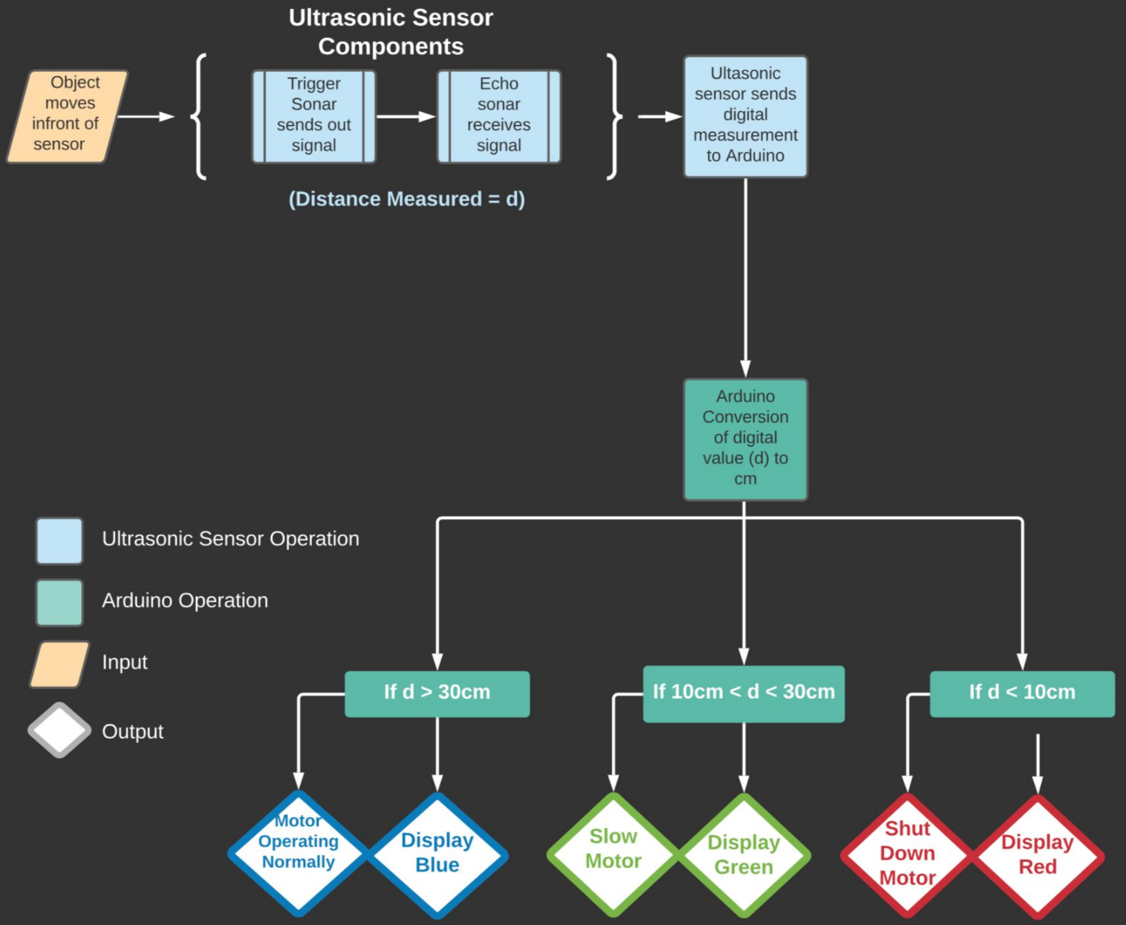
Lift Fan
Creates the Pressure underneath

CIRCUIT SCHEMATIC



_E550+1&934↓15∅16Φ3+84-/3165

FLOW CHART


Ultrasonic Sensor Components
Object moves infront of sensor
Trigger Sonar sends out signal
Echo sonar receives signal
(Distance Measured = d)
Ultasonic sensor sends digital measurement to Arduino
Arduino Conversion of digital value (d) to cm
Ultrasonic Sensor Operation
Arduino Operation
Input
Output
If d > 30cm
If 10cm < d < 30cm
If d < 10cm
Motor Operating Normally
Display Blue
Slow Motor
Display Green
Shut Down Motor
Display Red

_E550+1&934↓15∅16ф3+84-/3165

APPLICATIONS

HOVERCRAFT

Exploring braking technique

AUTONOMOUS VEHICLES

Autobraking safety measures

1Φ6850+1226Ø↓10∩5169640□3165

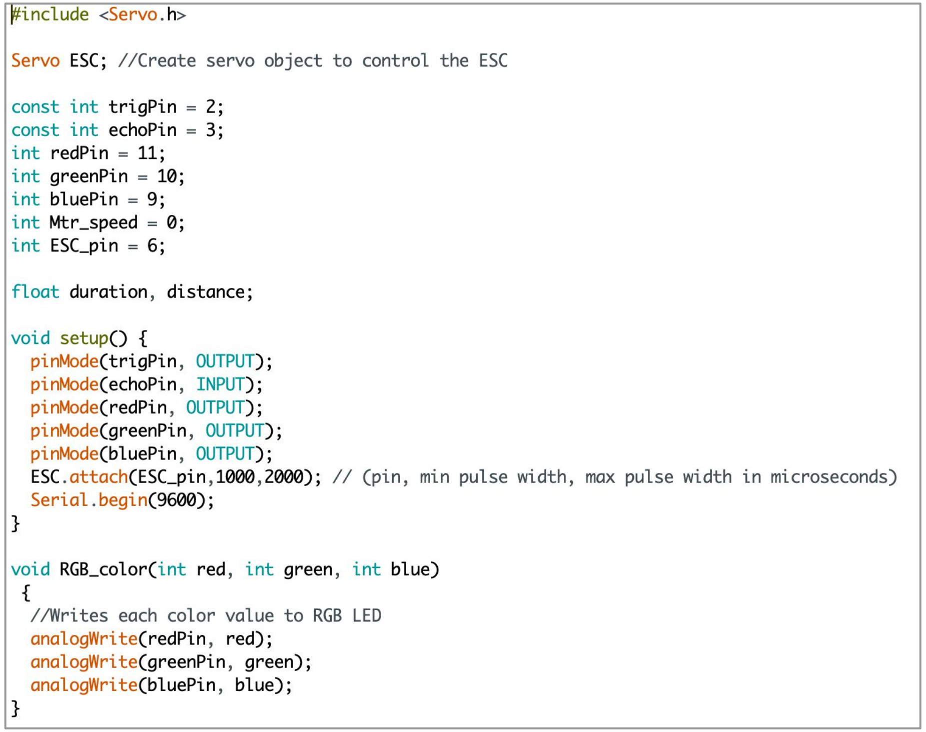
“I don’t have a fear of flying, I have a fear of crashing”

—Billy Bob Thornton

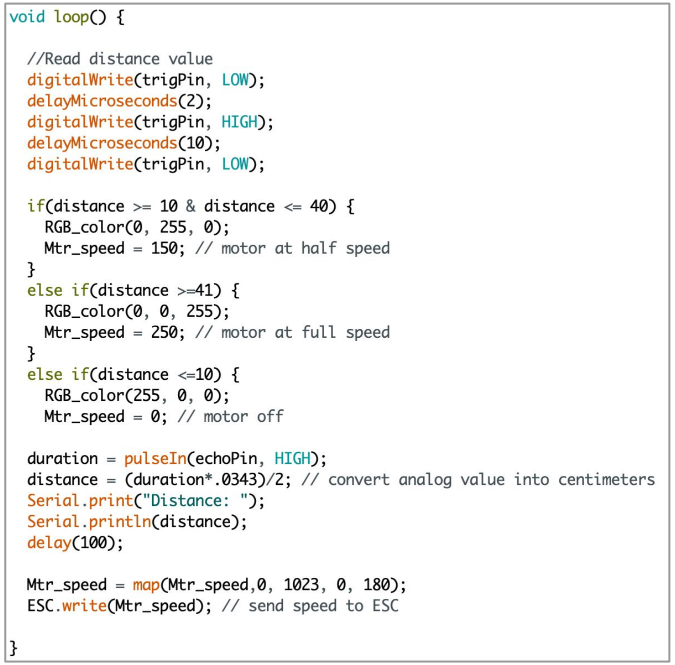
1960□749+€260⊓t-505∅+9/P+316

Appendix

Code



```
#include <Servo.h>

Servo ESC; //Create servo object to control the ESC

const int trigPin = 2;
const int echoPin = 3;
int redPin = 11;
int greenPin = 10;
int bluePin = 9;
int Mtr_speed = 0;
int ESC_pin = 6;

float duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  ESC.attach(ESC_pin,1000,2000); // (pin, min pulse width, max pulse width in microseconds)
  Serial.begin(9600);
}

void RGB_color(int red, int green, int blue)
 {
  //Writes each color value to RGB LED
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```



```
void loop() {

  //Read distance value
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  if(distance >= 10 & distance <= 40) {
    RGB_color(0, 255, 0);
    Mtr_speed = 150; // motor at half speed
  }
  else if(distance >=41) {
    RGB_color(0, 0, 255);
    Mtr_speed = 250; // motor at full speed
  }
  else if(distance <=10) {
    RGB_color(255, 0, 0);
    Mtr_speed = 0; // motor off

  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2; // convert analog value into centimeters
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);

  Mtr_speed = map(Mtr_speed,0, 1023, 0, 180);
  ESC.write(Mtr_speed); // send speed to ESC

}
```