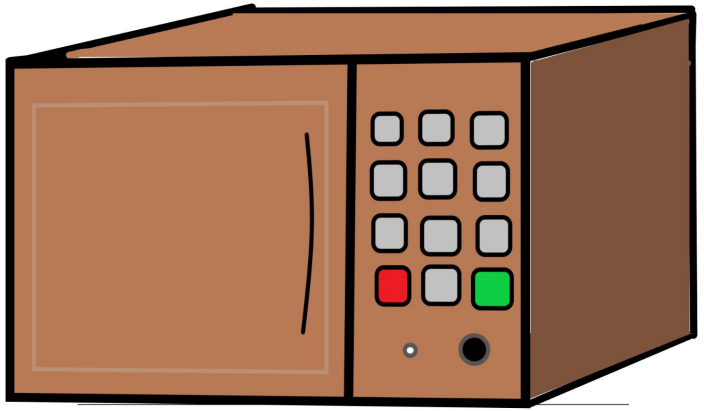
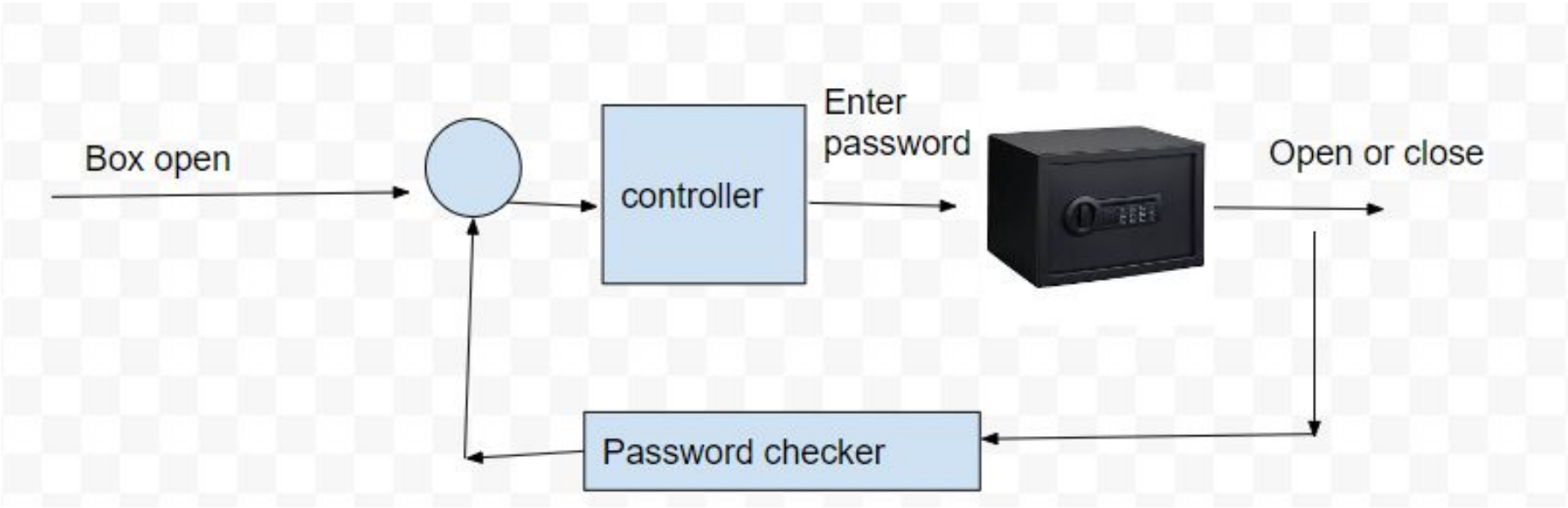
Arduino Final Projects:

Group 1:Ethan Aditya Anuj Tanay and Jonathan

Our group made a safe that stores things in a secure box that only people who know the code can get into. We used buttons to make a combination lock on the safe. To do that, in our code we made each button a variable and then when a code is entered it is verified using a loop that returns input. If the user messes up the code three times, the safe will lock and the buzzer will sound, it will wait ten seconds before everything resets and the user can try again.



Mechatronic design



Steps and components required to make our safe

Household items

- Cardboard
- Electrical tape
- Scissors

Arduino components/hardware

- Buttons
- Jumper cables
- Breadboard
- LED’s
- Buzzer
- Ultrasonic sensor
- Servos
- RGB lights

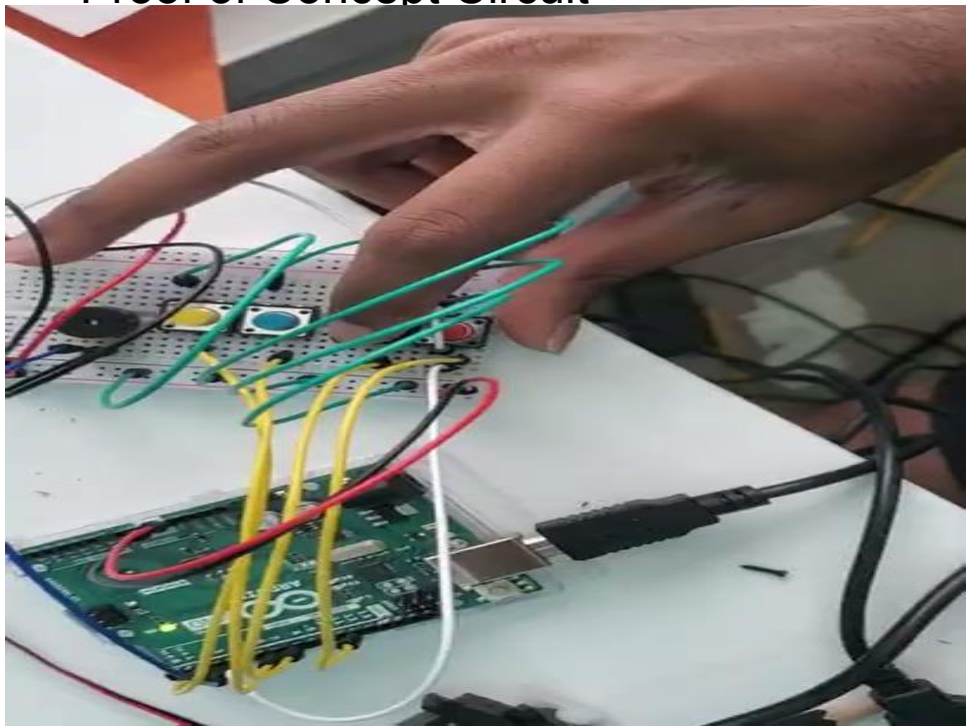
Instructions

1. We created a design plan for our project.
2. We created a drawing to help visualize how our project would like in real life.
3. We created code that we thought would work on the safe
4. We then built the safe using the materials listed on the left
5. We uploaded our code onto the Arduino connected to the safe.
6. We tested our project and fixed all of the problems that occurred in the safe and code

Results

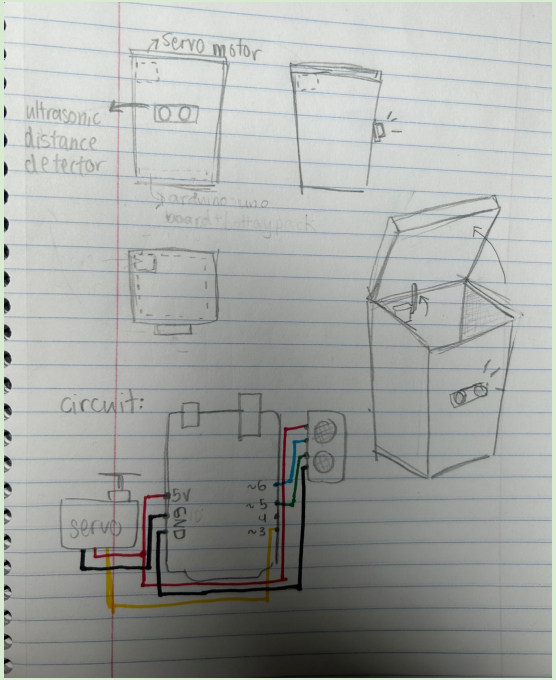
Actual Safe

Proof of Concept Circuit



Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

Automatic Trash Can Opener



Initial Design Drawing by Marianne

Description: Trash can that automatically opens when it senses that someone has walked up to it, and closes when the person walks away. A more sanitary, convenient, and accessible design.

Household Items:

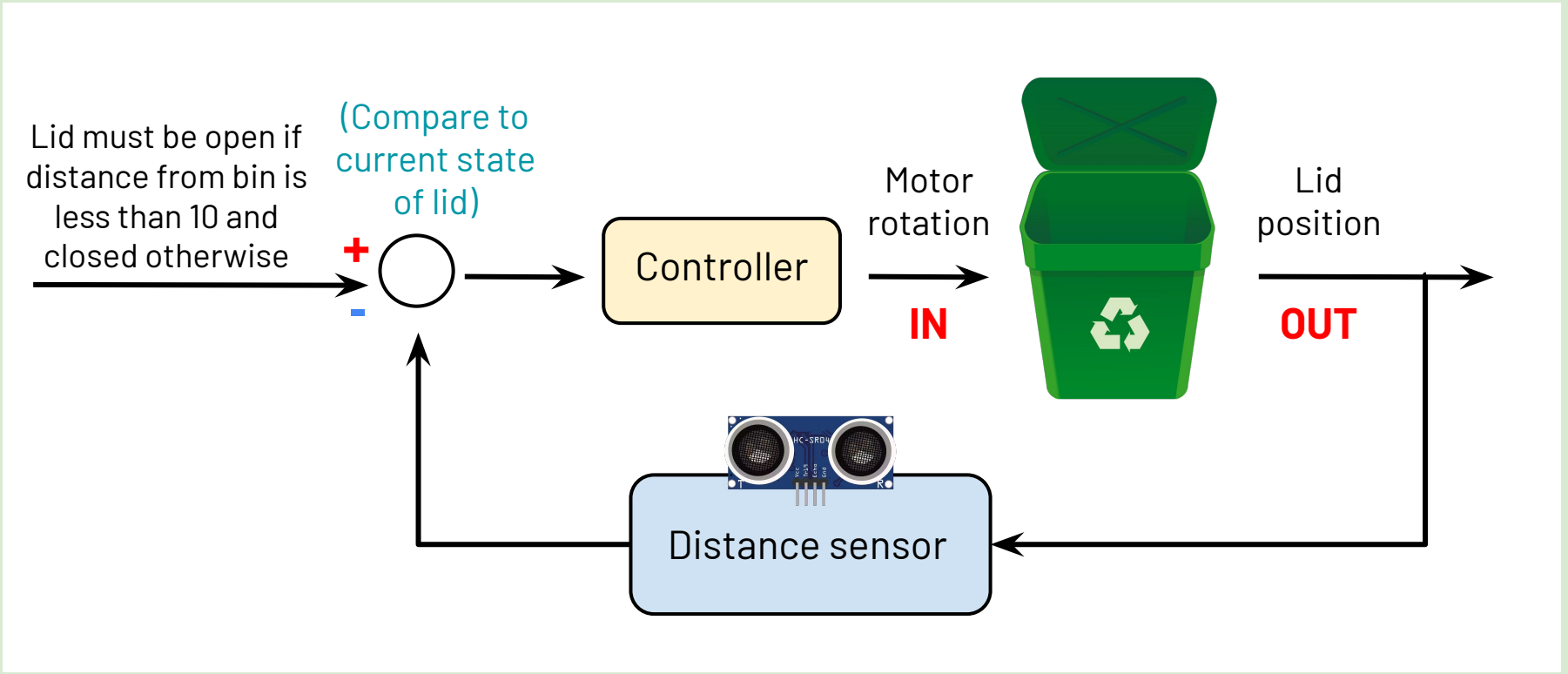
- Trash can
- Trash can lid
- Axle to connect Servo motor to hinge of trash can
- Cardboard

Arduino Components:

- Servo motor and arm attachment
- Ultrasonic sensor
- Jumper wires
- Switch
- 330 Ohm resistors

Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

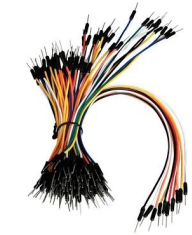
Control System Diagram- Closed Loop

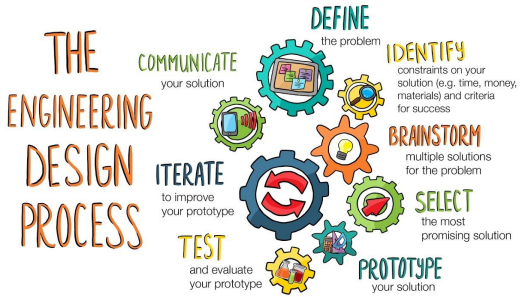


Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

Our Plan

1. Sketch drawing of design
2. Collect required Arduino components
3. Connect components and wires to breadboard
4. Attach servo motor to axle of trash can lid
5. Write and upload code
6. Test design and note down any problems with the code





Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

Code Process

We repurposed the code from the “Motion Alarm” activity, making several adjustments to it:

- Removed piezo buzzer and RGB LED
- Increased distance threshold so that it only opens when someone is nearby
- Changed servo motor rotation value in the if-else statement so that the lid would be at 90º when someone is nearby and 0º when no one is
- Added switch so that the system can be turned OFF or ON

Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

Sar's Code- Setup and Loop

```
*/

#include <Servo.h>                //include the servo library

const int trigPin = 11;           //connects to the trigger pin on the distance sensor
const int echoPin = 12;           //connects to the echo pin on the distance sensor
int switchPin = 7;                //switch to turn the system  on and off

float distance = 0;               //stores the distance measured by the distance sensor

Servo myservo;                    //create a servo object

void setup()
{
  Serial.begin (9600);        //set up a serial connection with the computer

  pinMode(trigPin, OUTPUT);   //the trigger pin will output pulses of electricity
  pinMode(echoPin, INPUT);    //the echo pin will measure the duration of pulses coming
  pinMode(switchPin, INPUT_PULLUP);   //set this as a pullup to sense whether the switch

  myservo.attach(9);            //use pin 9 to control the servo

}
```

```
void loop() {
  distance = getDistance();   //variable to store the distance measured by the sensor

  Serial.print(distance);     //print the distance that was measured
  Serial.println(" in");      //print units after the distance

  if (digitalRead(switchPin) == LOW) { //if the on switch is flipped
    if (distance <= 10) {                       //if a person is close

      //rotate servo to open the lid
      myservo.write(0);            //move the servo to 90 degrees


    } else {                                    //if a person is not close

      //rotate servo to close lid
      myservo.write(90);            //move the servo to 0 degrees

    }
  }

  delay(50);      //delay 50ms between each reading
}
```

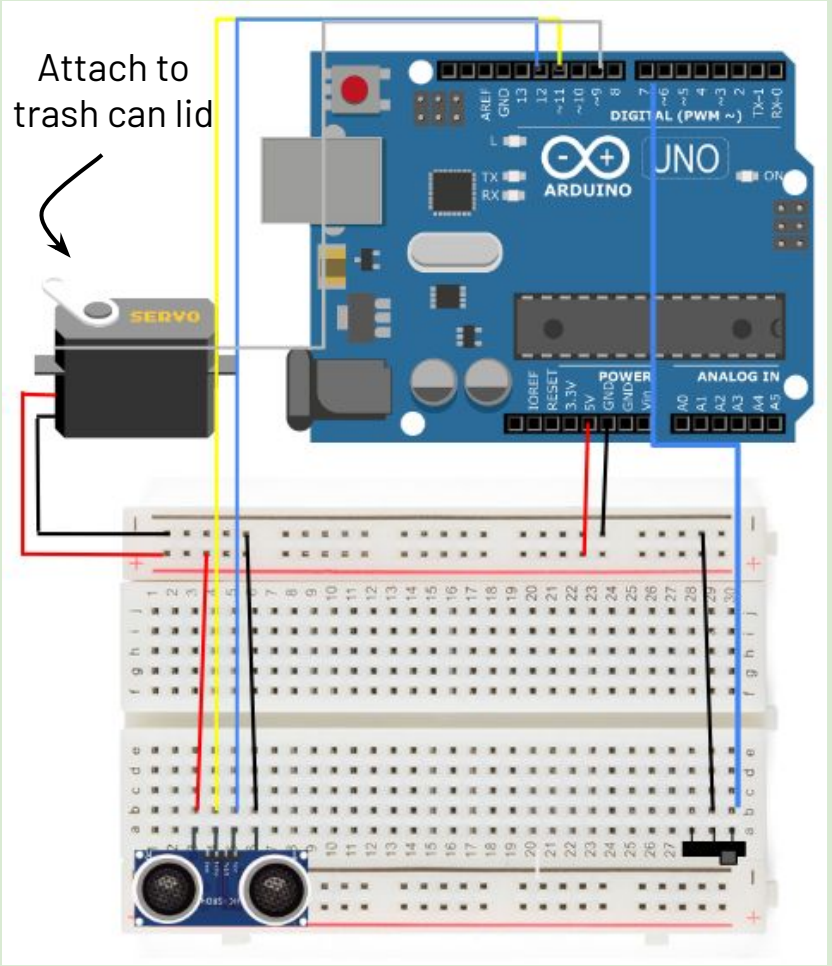
Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

Sar's Code- Function for Ultrasonic Sensor

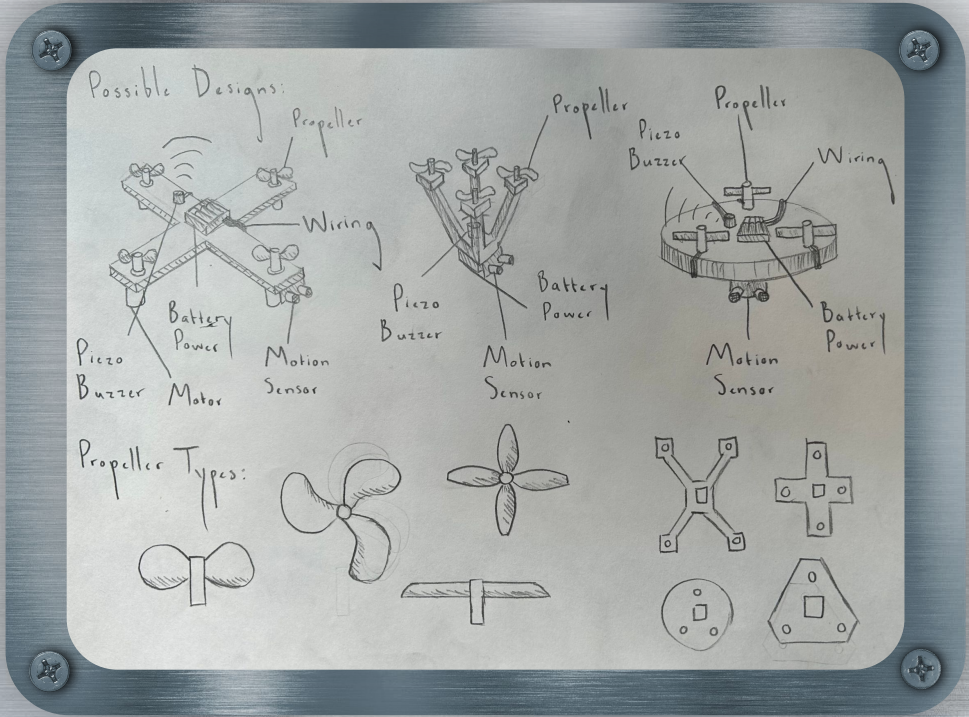
```
//------------------FUNCTIONS------------------------------

//RETURNS THE DISTANCE MEASURED BY THE HC-SR04 DISTANCE SENSOR
float getDistance()
{
  float echoTime;                   //variable to store the time it takes for a ping to bounce off an object
  float calculatedDistance;         //variable to store the distance calculated from the echo time

  //send out an ultrasonic pulse that's 10ms long
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  echoTime = pulseIn(echoPin, HIGH);      //use the pulsein command to see how long it takes for the
                                          //pulse to bounce back to the sensor

  calculatedDistance = echoTime / 148.0;  //calculate the distance of the object that reflected the pulse
                                          //(half the bounce time multiplied by the speed of sound)

  return calculatedDistance;              //send back the distance that was calculated
}
```

Group 2: Sarvagna, Marianne, Sofia, Mayah, Mizan

Sar's Version- Video & Schematic



Noah Drone



Arduino Components:

Motors
Motor Drivers
Piezo Buzzer
Motion Sensor
Wires

Household Items:

3D-Printed Base
3D-Printed Propellers
9 Volt Battery
Duct Tape

Meet the Team

Drew

(Production Designer)

Assembled drone

3D printed parts

Programming

Shrenik

(CAD Design Manager)

Designed drone body

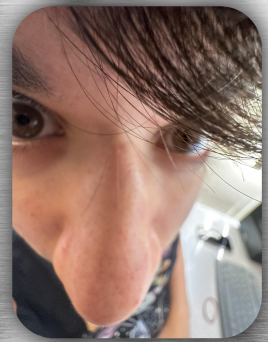
Conducted stress test

Max

(Documentation Expert)

Created visuals of preliminary designs

Designed slideshow

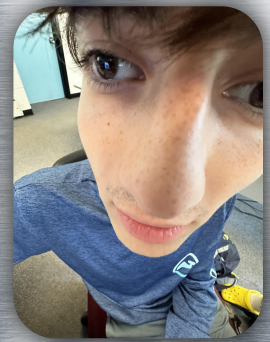


Bijan

(Secretary)

Created propellers

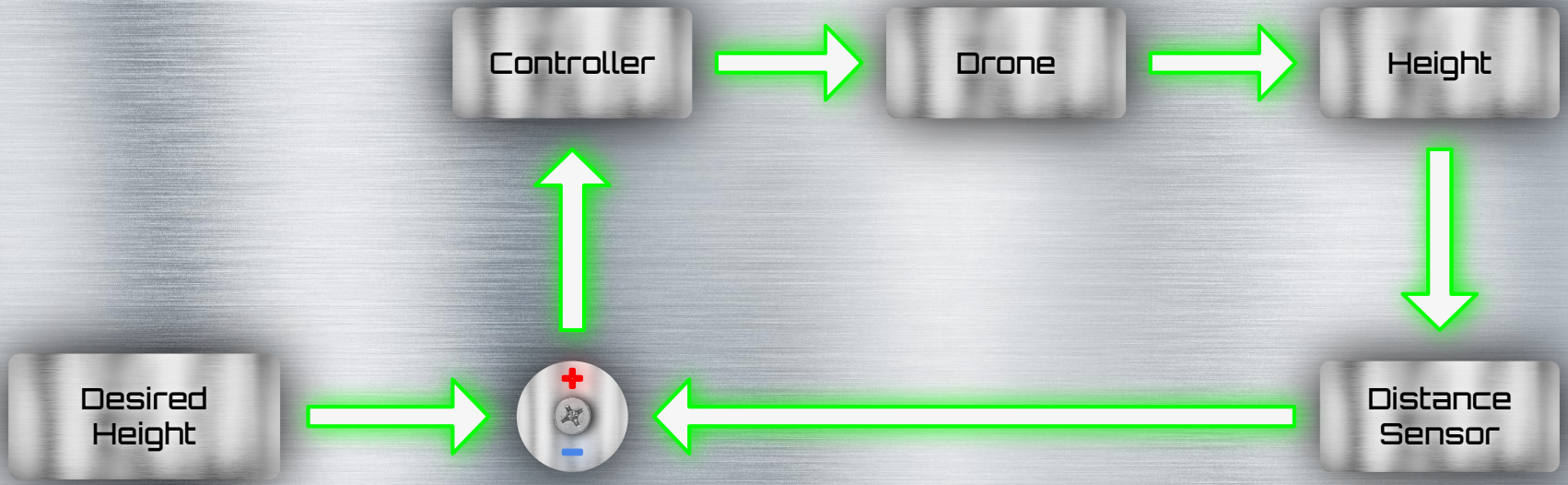
Created a replica of the motors on CAD

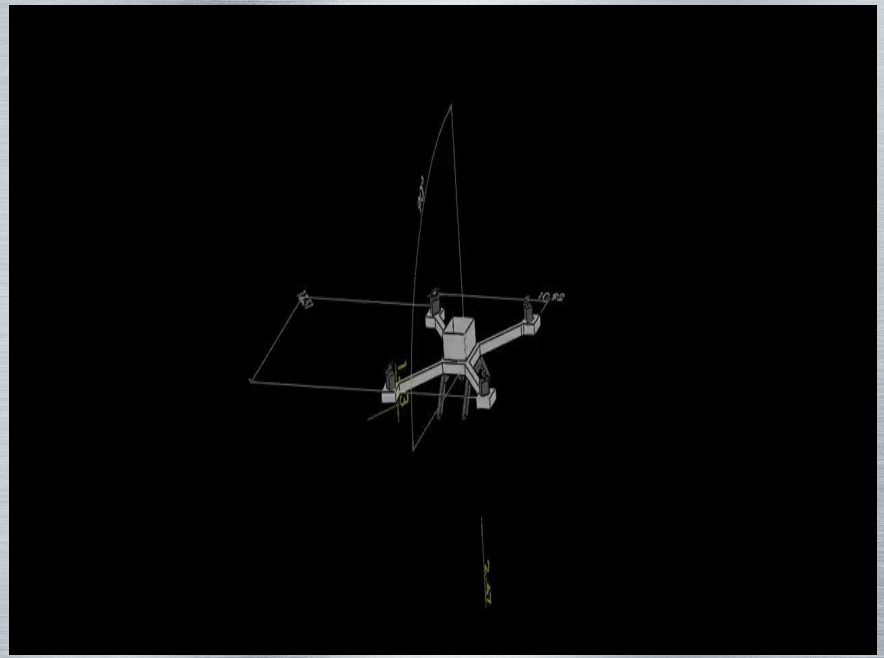


Group 3: Bijan, Drew, Max, and Shrenik

Our Noah Drone has two main purposes: carrying objects and detecting people or objects in range of the motion sensor. We first started the process of making our drone by creating a motor rendering on Solidworks; next, we found the dimensions of all the components that we required for the drone, such as the motion sensor, circuit board, and propellers. Once everything was properly measured, we started making a drone body design that would both hold all the components and provide enough room for each propeller. When we finished the original design of the body, we started designing the propellers. After everything was designed and put through stress tests, we printed each component and assembled the final drone, including the proper wiring on the circuit board.

Input-Output Diagram



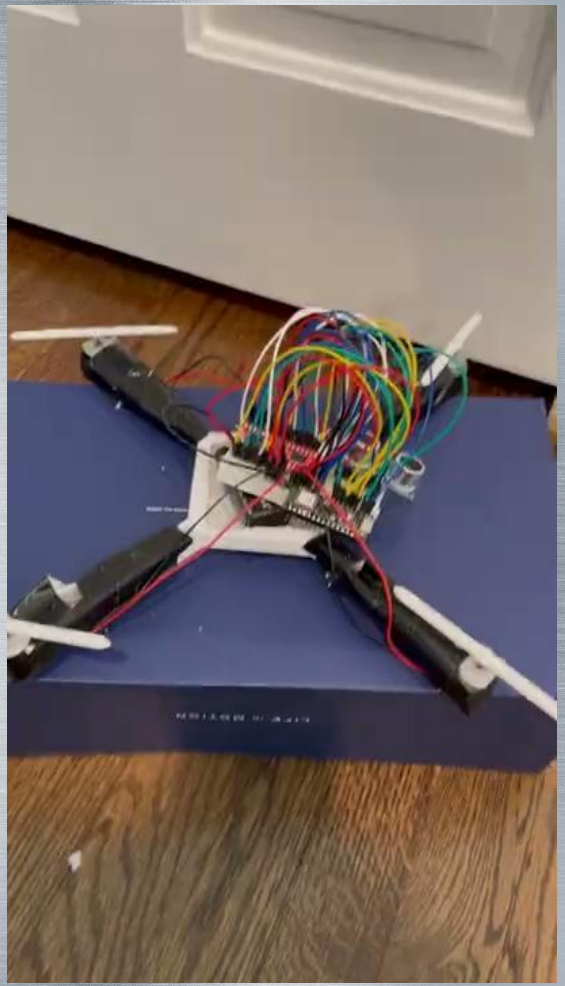


Assembly Video

Code For Flight

```
float dist;

void setup() {

void loop() {
 dist = getDist();

 if (dist < 8) {
   lift();
 }
 else if (dist == 8) {
   hover();
 }
 else {
   drop();
 }
}
void lift() {
 analogWrite(14, 255);
 analogWrite(15, 255);
 analogWrite(16, 255);
 analogWrite(17, 255);
}
void hover() {
 analogWrite(14, 128);
 analogWrite(15, 128);
 analogWrite(16, 128);
 analogWrite(17, 128);
}
```

```
void drop {
 analogWrite(14, 0);
 analogWrite(15, 0);
 analogWrite(16, 0);
 analogWrite(17, 0);
}
float getDist() {
 digitalWrite(13, HIGH);
 delayMicroseconds(10);
 digitalWrite(13, LOW);

 return(pulseIn(18,HIGH) / 148.0 );
}
```



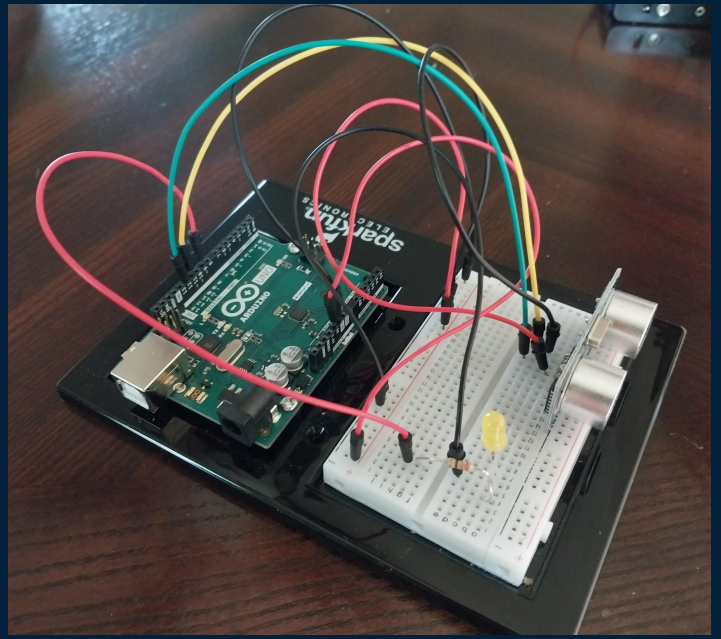
Noah Drone At Work

Creation and Results

Our Problem + Solution: (Group 4: Sahasra, Camila, Leah, and Shruthi)

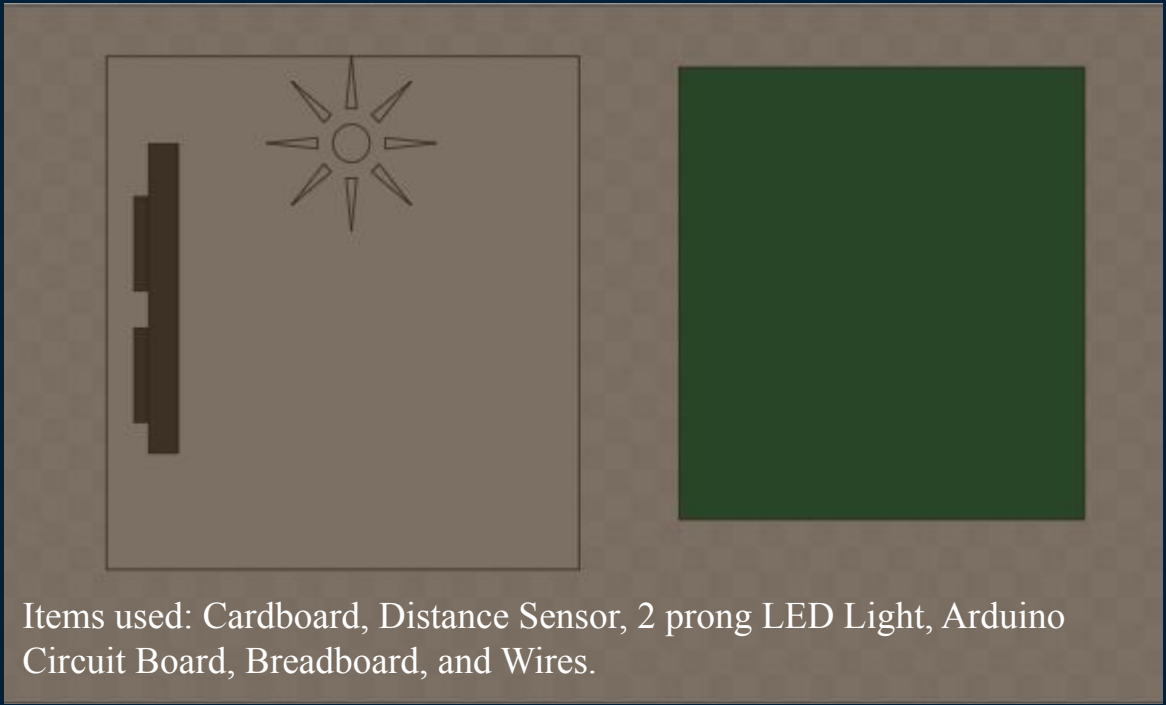
Our Problem: A common issue faced in many households is that people forget to turn off the lights after they leave a room, which wastes energy and money.

Our solution: Our project consists of a distance sensor, a 2 prong LED light, wires, and a 330 Ohm resistor. The distance sensor will be able to detect whether or not a person is in the room. If, after 1 minute of “scanning” the room, the sensor does not detect motion, it will trigger the LED light to turn off. Otherwise, the LED light will remain on.



Initial Design Drawing + Household Object + Code: (Group 4: Sahasra, Camila, Leah, and Shruthi)

Our prototype (includes only the distance sensor and 2 prong LED Light):

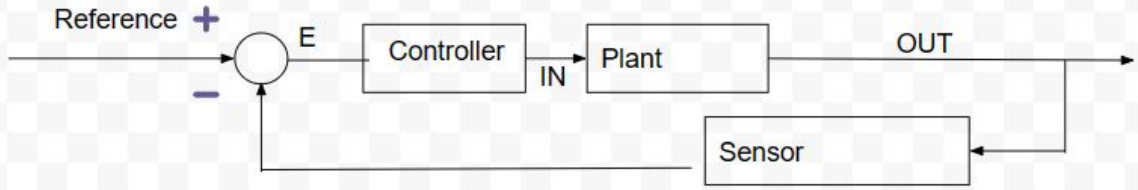


Items used: Cardboard, Distance Sensor, 2 prong LED Light, Arduino Circuit Board, Breadboard, and Wires.

Our household object:
The Arduino Circuit Board and breadboard will be placed inside a cardboard box. This box has a tiny hole for the LED Light to stick out.

Our Code:
We based our code off of the motion alarm project and the LED Light project and made many alterations for it to function for this particular use.

Closed Loop Control System Drawing: (Group 4: Sahasra, Camila, Leah, and Shruthi)



Reference: If the distance sensor doesn't detect motion, the LED Light should turn off. Otherwise, the LED Light should remain on.

Controller: Arduino Software
Plant: LED Light
Sensor: Distance Sensor

Inputs (IN):
- Whether or not the distance sensor is detecting motion
- Whether or not the LED Light is turned on

Outputs (OUT):
Plant: The brightness of the LED Light may change.
- Error (E): If the distance sensor detects motion, the LED Light will remain on.
- Error (E): If the distance sensor doesn't detect motion, it will trigger the LED Light to turn off.

Steps For Our Project: (Group 4: Sahasra, Camila, Leah, and Shruthi)

Steps:

1. Drew and finalized our sketch for our project
2. Attached the distance sensor, 2 prong LED light, 330 Ohm resistor, and wires to the Arduino circuit board and breadboard
3. Utilized sections of the code used for the “Motion Alarm” project and “Blinking LED” project to write our code
4. Tested our code multiple times and made several changes
5. Used a shoebox to house the circuit board and breadboard
6. Made a hole the size of the LED light on the top of the box
7. Made a hole in the side of the box, so the distance sensor is able to observe the surrounding area
8. Made a hole in the side of the box, so that we can connect the batteries to the circuit
9. Taped the circuit board and breadboard to the top of the cardboard box
10. Pulled the LED light through the hole
11. Connected the batteries to the circuit board

Pictures of our Code: (Group 4: Sahasra, Camila, Leah, and Shruthi)

```
// Include the necessary libraries


// Define the ultrasonic sensor pins
const int trigPin = 12;
const int echoPin = 11;


// Define the LED pin
const int ledPin = 9;


// Define the distance threshold for turning on the LED (in inches)
const int triggerDistance = 5;


// Define the time interval for gradually dimming the LED (in milliseconds)
const int dimInterval = 50;


// Define variables for LED control and timing
int ledBrightness = 0;
unsigned long ledOnTime = 0;
float distance = 0;
// Initialize the Ultrasonic sensor
```

Explanation: Defining the ultrasonic sensor pins, LED pin, trigger distance, dim interval, and other components

- The value of the “Trigger distance” refers to the maximum range that the distance sensor can detect objects.
- The “Dim Interval” refers to how fast the LED light will dim if motion hasn’t been detected after a minute.
- The “ledOnTime” is the timestamp of when the LED light was last turned on.

Pictures of our Code: (Group 4: Sahasra, Camila, Leah, and Shruthi)

```
void setup() {
// Set the LED pin as an output
pinMode(ledPin, OUTPUT);
pinMode(trigPin, OUTPUT); //this pin will send ultrasonic pulses out from the distance sensor
pinMode(echoPin, INPUT); //this pin will sense when the pulses reflect back to the distance sensor
Serial.begin(9600); //begin serial communication with the computer
Serial.print("To infinity and beyond!"); //test the serial connection
}


void loop() {
// Read the distance from the ultrasonic sensor in inches
//DETECT THE DISTANCE READ BY THE DISTANCE SENSOR
distance = getDistance();


Serial.print(distance);
Serial.println(" in"); // print the units
```

Explanation:

- Sending ultrasonic pulses from the distance sensor
- The “echoPin” will sense whether the pulses “reflect back”. If the pulses reflect back, then an object has been detected.
- Recording the distance from the distance sensor to the object

Pictures of our Code: (Group 4: Sahasra, Camila, Leah, and Shruthi)

```
// Check if something is closer than the trigger distance
if (distance < triggerDistance) {
// Restart the timer and turn on the LED
ledOnTime = millis();
ledBrightness = 255;
analogWrite(ledPin, ledBrightness);
}
```

```
// Check if the timer has elapsed (60 seconds)
if (millis() - ledOnTime >= 60000) {
// Gradually dim the LED
if (ledBrightness > 0) {
ledBrightness -= 5;
analogWrite(ledPin, ledBrightness);
delay(dimInterval);
}
}
delay(50);
}
```

Explanation:

- Checking whether the distance to the object is within 5 inches. If so, then the LED Light will turn on and this new timestamp will be recorded.
- If the time that has passed since the LED light has been turned on is greater than 6000 milliseconds (or 60 seconds), the LED light will start to dim in increments of 5.

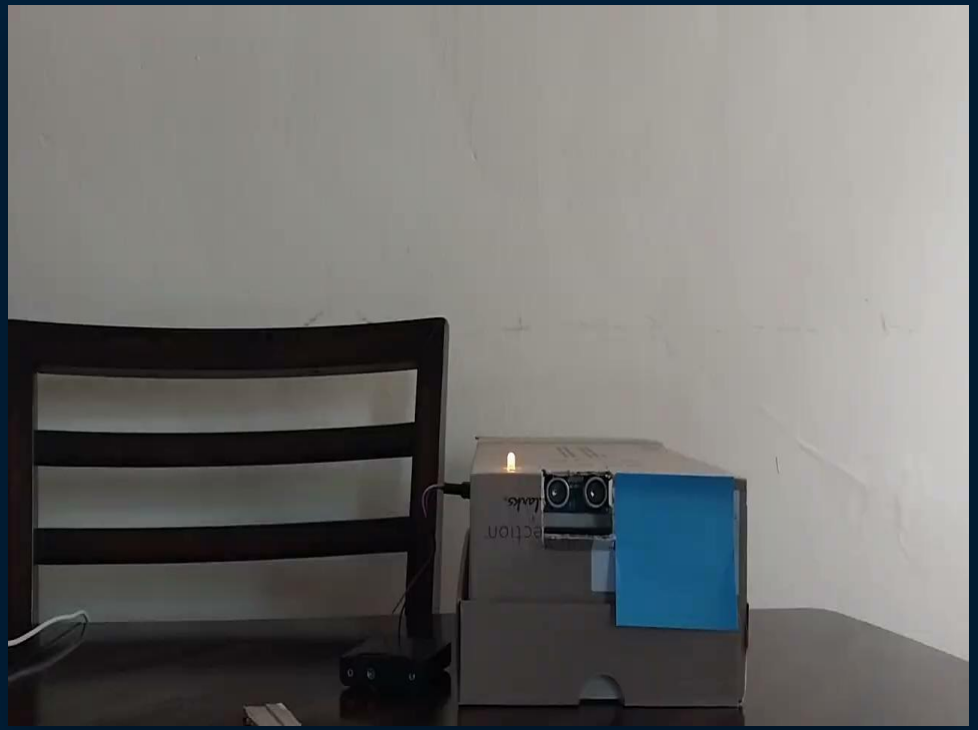
Pictures of our Code: (Group 4: Sahasra, Camila, Leah, and Shruthi)

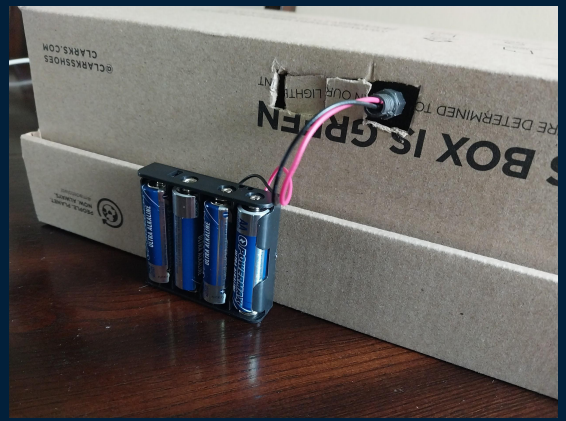
```
//send out an ultrasonic pulse that's 10ms long
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);


echoTime = pulseIn(echoPin, HIGH); //use the pulsein command to see how long it takes for the
//pulse to bounce back to the sensor


calculatedDistance = echoTime / 148.0; //calculate the distance of the object that reflected the pulse (half the bounce time multiplied by the speed of sound)


return calculatedDistance; //send back the distance that was calculated
}
```

Explanation:

- Calculates the distance from the distance sensor to the object by dividing how long it takes for the pulse from the distance sensor to bounce back by the speed of sound in the air (148.0)

Photos + Video: (Group 4: Sahasra, Camila, Leah, and Shruthi)





Our Observations + Tasks: (Group 4: Sahasra, Camila, Leah, and Shruthi)

Observations:

We noticed that the distance sensors are pretty small and will only be able to observe certain areas in the room. In order to implement our idea into houses and other buildings, we'll need to use bigger distance sensors. This means that we'll need to utilize larger circuits, wires, 2 prong LED lights, breadboards, and cardboard boxes.

While designing our final project, we realized that the shape and structure of rooms varies. Therefore, some particular rooms may need more than one distance sensor in order to observe every square inch of the room. In this case, we may need multiple breadboards, circuits, and cardboard boxes. Additionally, for these special instances, we may need to modify our code in order to take all of the distance sensors' observations into consideration; this will minimize the number of times that our circuits make errors and will make our project for precise.

Tasks:

Sahasra- Presentation

Leah- Arduino Board and Code

Camila- Arduino Board and Code

Shruthi- Presentation

Group 5: Samartha, Aarav, Pranav

Problem: The temperature in the summer is uncomfortable to many people, and many people do not have functioning A/C's and fans in their rooms.

The Solution: Due to the heat of summer it is normal to want a **portable fan** that *doesn't require a lot of energy* to operate. It also saves power by only turning on when a person(detected by the ultrasonic sensor) is nearby.

Group 5: Samartha, Aarav, Pranav - Procedure

How we created our prototype:

1. We took inspiration from the first motion sensor project on the guide slide with the servo motor.
2. We recreated the entire project just to check and observe again how the motor reacted when the code started and it sensed motion
3. We realized then that the led wasn't needed for what we were trying to make
4. We also realized that the initial motor didn't rotate 360° like we expected it to so we changed it to the yellow DC motor
5. We made the propeller for the fan out of cardboard to represent the household object we were including.
6. Finally, we changed the code to have the propeller turn on when an object was sensed

Group 5: Samartha, Aarav, Pranav - Challenges

Problem:

Due to the fan spinning very fast it was common for the propeller to fly off a number of times.

Solution:

We designed a cardboard ring to go around the DC motor which would secure the propeller and prevent it from flying off.

Group 5: Samartha, Aarav, Pranav - Code

```
// Include the necessary libraries


// Define the ultrasonic sensor pins
const int trigPin = 22;
const int echoPin = 23;


// Define the motor pin
const int motorPin = A0;


// Define the distance threshold for turning on the motor.
const int triggerDistance = 1;


unsigned long motorOnTime=0;


float distance = 0;
// Initialize the Ultrasonic sensor




void setup() {
// Set the LED pin as an output
pinMode(motorPin, OUTPUT);
pinMode(trigPin, OUTPUT); //this pin will send ultrasonic pulses out from the distance sensor
pinMode(echoPin, INPUT); //this pin will sense when the pulses reflect back to the distance sensor
Serial.begin(9600); //begin serial communication with the computer
Serial.print("To infinity and beyond!"); //test the serial connection
}

void loop() {
// Read the distance from the ultrasonic sensor in inches
//DETECT THE DISTANCE READ BY THE DISTANCE SENSOR
distance = getDistance();


Serial.print(distance);
Serial.println(" in"); // print the units
if (distance < triggerDistance) {
// Restart the timer and turn on the LED
motorOnTime = millis();
analogWrite(motorPin, 255);


}




// Check if the timer has elapsed (60 seconds)
if (millis() - motorOnTime >= 5000) {
analogWrite(motorPin,0);
}


delay(50);
}
```

Group 5: Samartha, Aarav, Pranav - Code

```
65
66
67
68
69  float getDistance()
70  {
71  float echoTime; //variable to store the time it takes for a ping to bounce off an object
72  float calculatedDistance; //variable to store the distance calculated from the echo time
73
74
75  //send out an ultrasonic pulse that's 10ms long
76  digitalWrite(trigPin, HIGH);
77  delayMicroseconds(10);
78  digitalWrite(trigPin, LOW);
79
80
81  echoTime = pulseIn(echoPin, HIGH); //use the pulsein command to see how long it takes for the
82  //pulse to bounce back to the sensor
83
84
85  calculatedDistance = echoTime / 148.0; //calculate the distance of the object that reflected the pulse (half the bounce time multiplied by the speed of sound)
86
87
88  return calculatedDistance; //send back the distance that was calculated
89  }
90
91
92
93
94
95
96
97
```

Group 5: Samartha, Aarav, Pranav - Image and Videos