Self Watering Plant

Group: 4

DES634: Electronics for Designers

Concept

A Self Watering Plant

Proper plant care is a fundamental aspect of maintaining healthy and thriving greenery, yet it often poses challenges to plant enthusiasts when they have other commitments to keep. In response to this, we have developed a solution that merges the beauty of nature with the power of technology: The self and smart watering plant device.

Acts as:

- An automated solution to irregular plant watering
- A perfect home-decor
- An Ambient White Nightlamp

Inspiration

Inspiration for the product came from problem of inconsistent watering. The idea was to build a product that only resolves this problem but also adds decor value to the place.
For multi-utilities of the product, several modes were added corresponding to the use case.
The two major factors that inspired us were:

Importance of Plant Health: Inconsistent watering, whether too much or too little, can have detrimental effects on plant health. Overwatering can lead to root rot, fungal diseases, and nutrient leaching, while underwatering can cause wilting, stunted growth, and even plant death. Providing plants with the optimal amount of water at the right time is crucial.

User Convenience: Not everyone has the time, knowledge, or inclination to monitor and adjust watering schedules regularly. Our product offers convenience and peace of mind to plant caregivers by automating the watering process. Whether they're busy professionals, frequent travelers, or simply lack a green thumb, users can rest assured that their plants are being cared for effectively.

Design

Our smart watering plant device is not only functional but also aesthetically pleasing, drawing inspiration from modern design trends and innovative technological solutions.

Functional Segregation:

- We adopted a design approach that separates the functional components of the device into distinct compartments for soil/plant and water.
- The top compartment houses the soil and plant, incorporating a soil moisture content sensor to monitor the plant's hydration needs.
- The bottom compartment stores the water reservoir and features a water level sensor to ensure optimal water supply.

Design

Efficient Automation:

- A motor mechanism is integrated into the device, triggered by the soil moisture content sensor. This automation ensures timely watering based on plant requirements, enhancing plant health and minimizing manual intervention.

Transparency and Visibility:

- Inspired by modern devices with transparent casings, we've designed our device to showcase the internal electronics.
- The base of our device consists of two layers: an inner casing housing all electronics and an outer layer providing support and aesthetics.
- This design allows users to visually appreciate the technology behind the smart watering plant while adding a contemporary touch to any environment.

Aesthetic Enhancement:

By diffusing light on the outer walls of the inner casing, the base creates a soft glow, adding a subtle yet captivating ambiance to any space

Conceptualisation

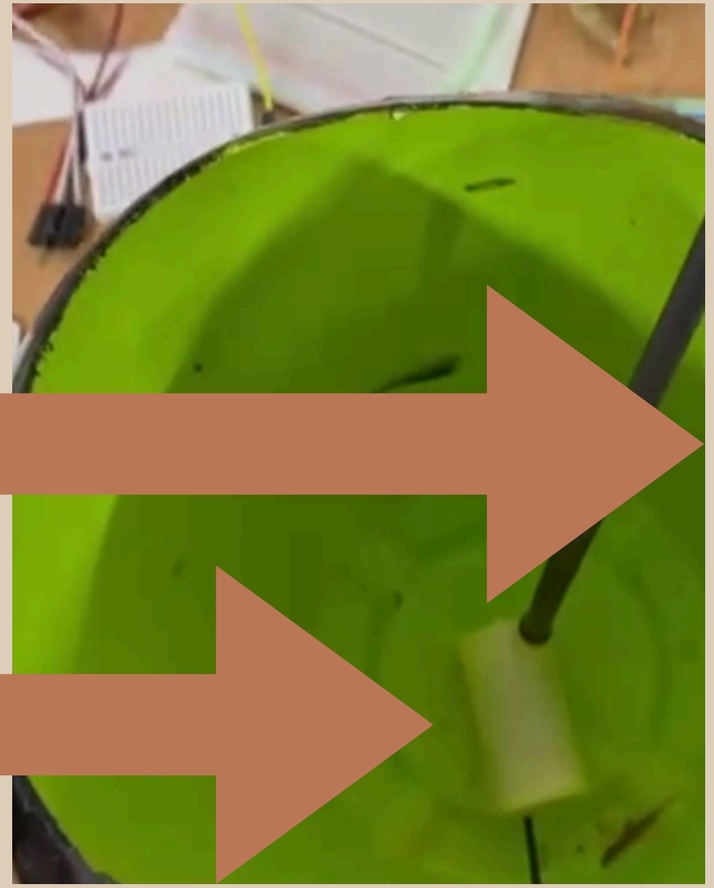
The design was made using already available pots and modifying them to suit our design needs.

Plant Compartment

Funnel for Water Refilling

Soil Moisture Content Sensor

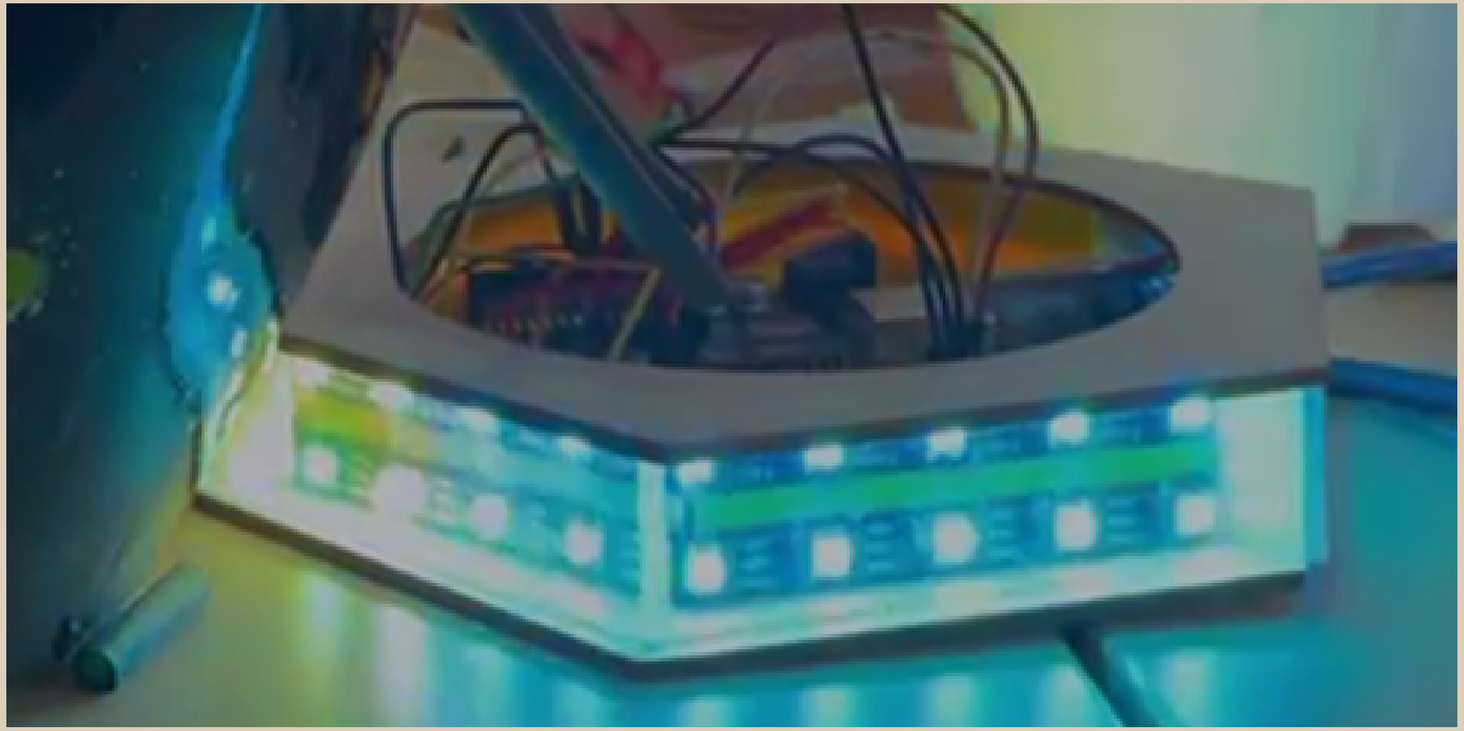
Water Compartment

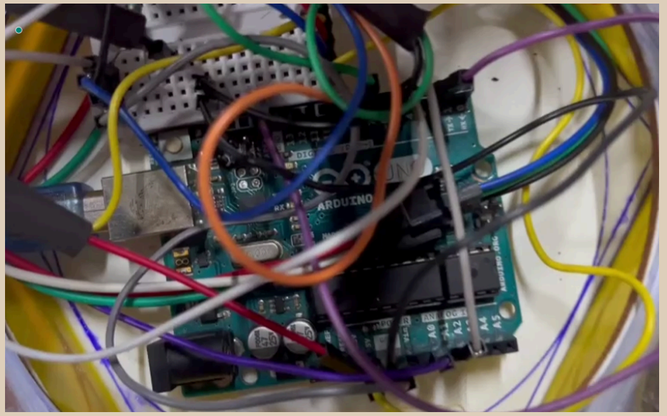


Water Level Sensor

Motor

Electronics Compartment





Componenets

Arduino UNO

Capacitive Soil Moisture Sensor

Submersible Mini Water Pump

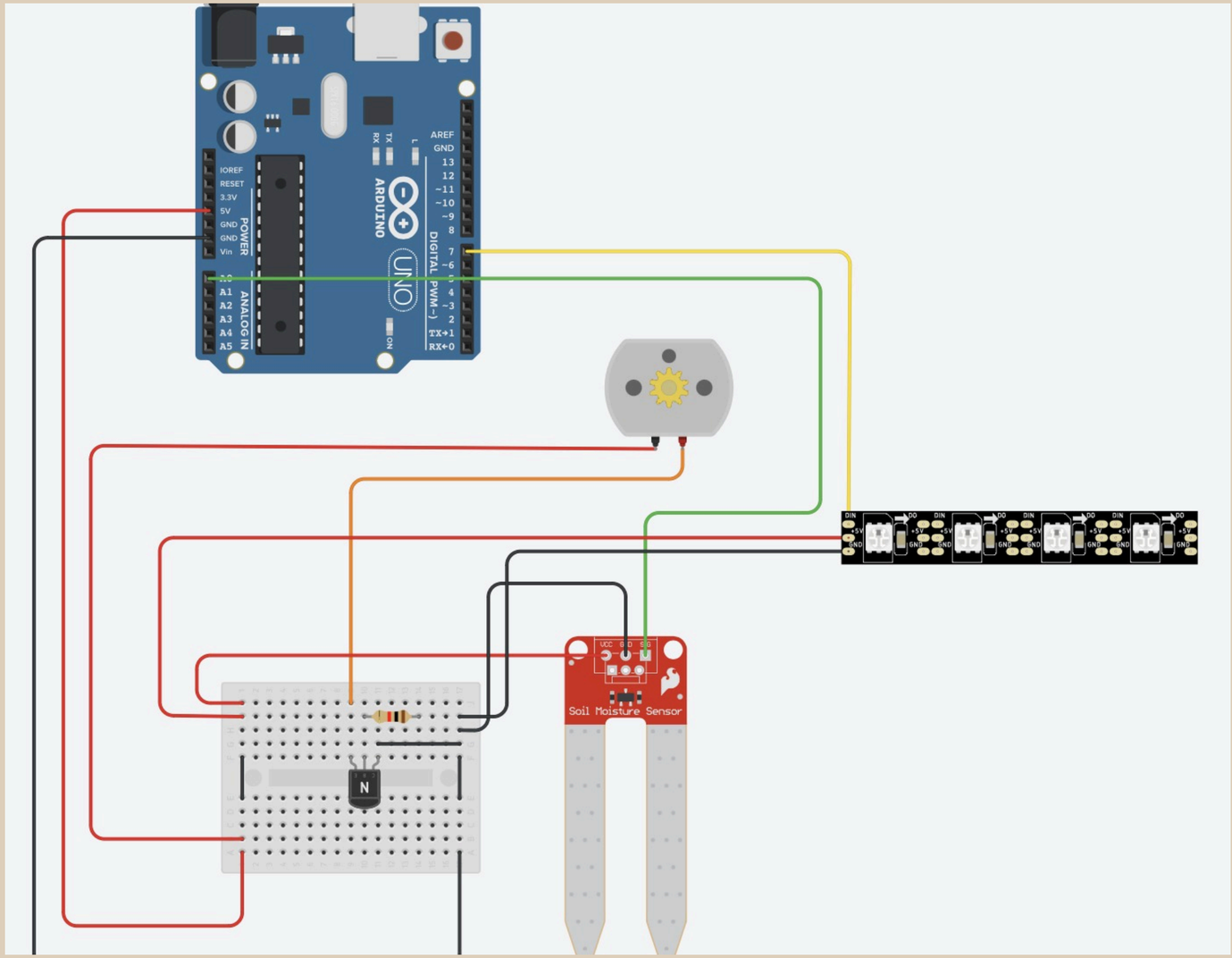
2N2222 NPN BJT Switching Transistor

Water Level Depth Detection Sensor Module

RGB Addressable LED Module

Jumper Wires

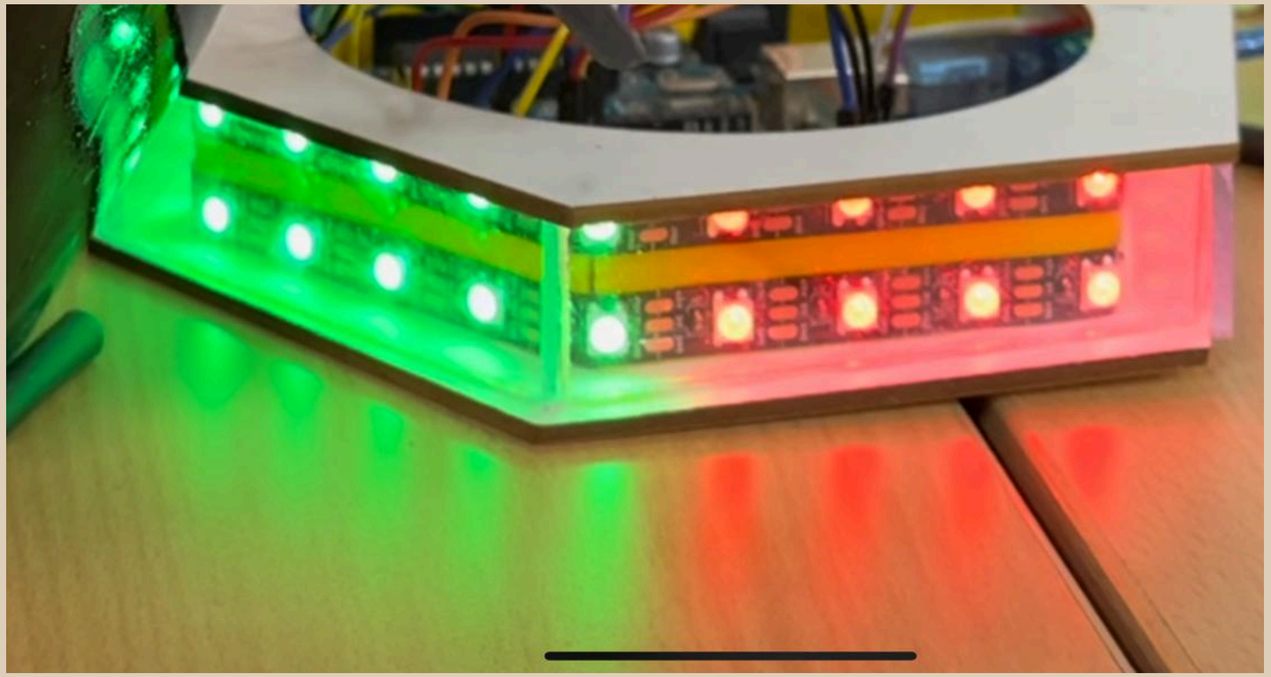
System Architecture



Modes

Moisture Content Indicator

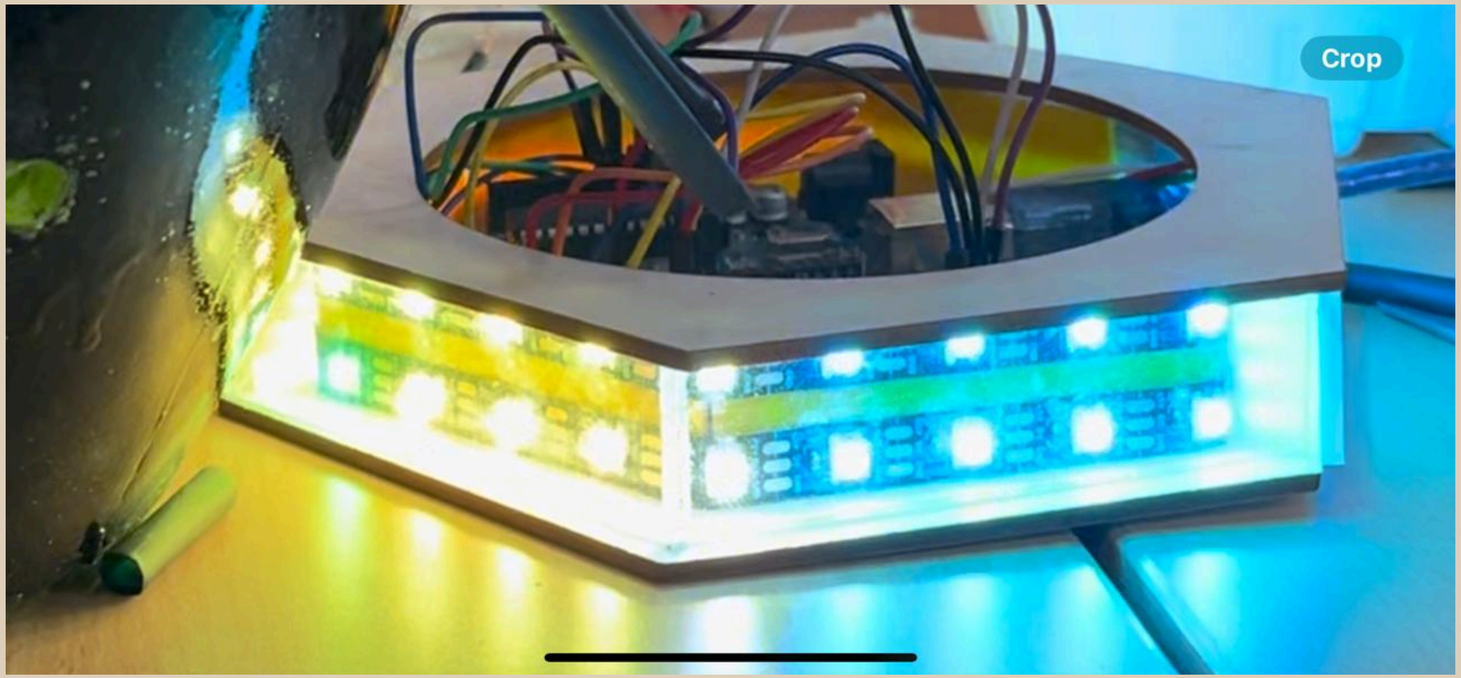
- The number of green lights scrolling around gives an idea of moisture content of soil.
- A completely green light signifies good water content and plant health while a red one corresponds to low moisture levels.
- The water level sensor when reaches a threshold starts blinking blue light irrespective of the mode to stop overflow.



Modes

Home Decor

- Lights can be used for Home Decor and adding lighting based on the mood and atmosphere for lively parties to calm afternoons.



Modes

Night Lamp

- The LEDs at the base can also be used as night lamp.
- It provides warm white light which soothes and provides a comfortable atmosphere for sleeping.

Code

```
#include <FastLED.h>
#define LED_PIN1  7
#define LED_PIN2  8
#define NUM_LEDS  29
int data;
int levelPin = A0;
int sensorPin = A2;
int motor = 4;
CRGB leds1[NUM_LEDS];
CRGB leds2[NUM_LEDS];
int bluetoothvalue=1 ;
int n=29;
int level;
int redcount=15;
void setup()
{
 pinMode(motor, OUTPUT);
 digitalWrite(3,HIGH);
 Serial.begin(9600);
 FastLED.addLeds<WS2812, LED_PIN1, GRB>(leds1, NUM_LEDS);
 FastLED.addLeds<WS2812, LED_PIN2, GRB>(leds2, NUM_LEDS);
 //pinMode(LED_BUILTIN, OUTPUT);
}
```

```
void loop()
{

level = analogRead (levelPin);

int moisture = analogRead(A2);
if(moisture > 565)
{
 Serial.println("hello");
 digitalWrite (motor, HIGH);
 delay(5000);}
else
{digitalWrite (motor, LOW);}

if(moisture>5650)
{redcount=29;}
else if(moisture>=530 && moisture <565)
{redcount = 20;}
else if (moisture>=500 && moisture <530)
{redcount=9;}
else if (moisture >=470 && moisture <500)
{redcount =0;}
else
{redcount =0;}
Serial.println(moisture);
```

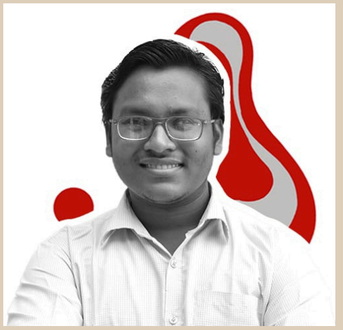
Code

```
if(level>=500)
{ for (int i = 0; i < n; i++) {
  leds1[i] = CRGB(0, 0, 265);
  leds2[i] = CRGB(0, 0, 265);
} delay(300); FastLED.show();
for (int i = 0; i < n; i++) {
  leds1[i] = CRGB(0, 0, 0);
  leds2[i] = CRGB(0, 0, 0);
}delay(300);
FastLED.show();
}

// Turn half of the LEDs red
else if(bluetoothvalue==1)
{
for (int i = 0; i < redcount; i++) {
  leds1[i] = CRGB(265, 0, 0);
  leds2[i] = CRGB(265, 0, 0);
}
// Turn the other half blue
for (int i = redcount; i < n; i++) {
  leds1[i] = CRGB(0, 265, 0);
  leds2[i] = CRGB(0, 265, 0);
}

// Display the LEDs
FastLED.show();

// Shift the LEDs to create the circling effect
for (int j = 0; j < n; j++) {
  // Shift the LEDs by one position to the right
  for (int i = n - 1; i > 0; i--) {
    leds1[i] = leds1[i - 1];
    leds2[i] = leds2[i - 1];
  }
  // Move the first LED to the last position
  leds1[0] = leds1[n - 1];
  leds2[0] = leds2[n - 1];


  // Display the shifted LEDs
  FastLED.show();


  // Delay to control the speed of the circling effect
  delay(55); // Adjust the delay time as needed
 }

}

else if(bluetoothvalue==2)
{
int half = n / 2; // assuming n is even

// Turn half of the LEDs red
for (int i = 0; i < half; i++) {
  leds1[i] = CRGB(100, 100, 0);
  leds2[i] = CRGB(100, 100, 0);
}
// Turn the other half blue
for (int i = half; i < n; i++) {
  leds1[i] = CRGB(0, 100, 100);
  leds2[i] = CRGB(0, 100, 100);
}

// Display the LEDs
FastLED.show();

// Shift the LEDs to create the circling effect
for (int j = 0; j < n; j++) {
  // Shift the LEDs by one position to the right
  for (int i = n - 1; i > 0; i--) {
    leds1[i] = leds1[i - 1];
    leds2[i] = leds2[i - 1];
  }
  // Move the first LED to the last position
  leds1[0] = leds1[n - 1];
  leds2[0] = leds2[n - 1];
  // Display the shifted LEDs
  FastLED.show();
  // Delay to control the speed of the circling effect
  delay(15); // Adjust the delay time as needed
}

}

else if(bluetoothvalue == 3)
{
 for (int i = 0; i < n; i++) {
 leds1[i] = CRGB(246, 231, 210);
 leds2[i] = CRGB(246, 231, 210);
 FastLED.show();
}

}

}
```

TEAM

Rishabh Raj
Electrical Engg
210846

Sanchit Hari
Civil Engg
210920



Kamal Sahu
M.Des
231190012

Contributions

Kamal Sahu: Fabrication, Form Development

Rishabh Raj: Conceptualisation, Form Development, Circuit Building & Video

Sanchit Hari: Conceptualisation, Fabrication, Form Development, Coding and Presentation

Thank You