IoT based Smart Waste Segregator

A PROJECT REPORT

submitted in partial fulfillment of the requirements

for the course

EEE2004 Measurement and Instrumentation

by

T Sakthi Ram 20BEE1026

Anirudh Menon 20BEE1100

Mruthunjay 20BEE1132

under the guidance of

Dr. S. Hemamalini

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRICAL ENGINEERING

VIT Chennai

December 2021

Link to the Updated Video:

https://drive.google.com/file/d/1G2s-Vowo7XgoTKom1lyu72p7eCUS7hUu/view?usp=sharing

Abstract:

Garbage segregation is the process of separating different types of waste before disposal. It vastly simplifies the recycling process. Effective segregation of wastes ensures that lesser waste goes to landfills which makes it cheaper and better for people and the environment. It is also beneficial to improve public health. In particular, hazardous wastes can cause long term health problems, so its really important that they are disposed off of correctly and safely and not mixed in with the normal waste from homes or offices. Majority of the waste fall under three categories namely dry waste, wet waste and metal waste. It becomes impossible to recycle waste and use it for generation of biogas or any useable products when unsegregated, making it an important step in proper waste disposal. We have also used IoT module to send the data to a cloud database from which the waste level at any point of time can be found which makes the progress of managing bins way more efficient.

Table of Contents

List of Tables

List of Figures

Contents

1.1 Introduction

Garbage sorting is the process of segregating waste into different categories such that is makes the process of either disposing them or recycling them more effective and easier. The most common categories of segregation is wet and dry waste. Wet waste generally contains organic waste which are damped. The segregation of waste into these two broad categories is really helpful because the wet waste can be used for composting while dry waste can be used for recycling.

1.1.1 Motivation

Pollution is the alarming issue of the future. Hence the efficient use of resources is the need of the hour. Everyday although there is progress in technology the waste disposal system in our country still follows primate practices. Controlling the waste not only helps in reduction of landfills but we could explore the possibilities of renewable energy from compostable waste and it would increase the feasibility of recycling the dry waste if it is segregated much earlier.

1.1.2 Objectives

Garbage segregation is the process of separating different types of waste before disposal. It vastly simplifies the recycling process.

Effective segregation of wastes ensures that lesser waste goes to landfills which makes it cheaper and better for people and the environment.

It is also beneficial to improve public health. In particular, hazardous wastes can cause long term health problems, so its really important that they are disposed off correctly and safely and not mixed in with the normal waste from homes or offices.

Majority of the waste fall under three categories namely dry waste, wet waste and metal waste. It becomes impossible to recycle waste and use it for generation of biogas or any useable products when unsegregated, making it an important step in proper waste disposal.

1.1.3 Scope of the work

The automatic waste segregator along with IoT based waste level detection when implemented in large scale it could help in automation and managing of waste segregation. In future we can further incorporate Machine learning to predict the expect usage of bins and manage the bins network accordingly

CHAPTER 2

2.0 Overview of Project

Our project has two primary functions

- *Automate Segregation of the waste into three bins i.e. dry and wet waste. We have achieved this with the use of soil moisture sensor which senses the wetness of the waste and digital capactive senor which senses the presence of waste and two servo motors which aids us in making the mechanism to put the waste in correct bin according its characteristiscs all these sensors are controlled by Arduino Uno*

- *Indicate the Filled Capacity in each bin using IOT and APP.*
  We have used Esp8266 IoT module and ultrasonic distance sensors to achieve this purpose. The ultrasonic distance sensors give the amount of space filled in each bin which is then uploaded to a cloud database known as firebase with the help of ESP module. The data from the cloud database can be accessed through our Custom made Mobile App.

2.1 Detection of Presence of waste

We don't want the device to be running all the time so we are using a digital capacitive touch sensor which will helps us to turn on the device whenever it feels some force of waste getting put on it

2.2 Detection of waste category

We can detect the type of waste by using the soil moisture sensor. The working of the soil moisture sensor is pretty straightforward. The fork-shaped probe with two exposed conductors, acts as a variable resistor. This resistance is inversely proportional to the soil moisture:

- *The more water in the soil means better conductivity and will result in a lower resistance.*
- *The less water in the soil means poor conductivity and will result in a higher resistance.*

The sensor produces an output voltage according to the resistance, which by measuring we can determine the moisture level. We can fix a certain minimum threshold of waste level above the waste will be considered wet and hence segregate into two categories.

2.3 Segregating the waste by using Servo motors

We have created a rotating platform which can rotate upto 180 degree . At rest it is set at 90 degree when the waste detected is dry it moves to 0 degree and when the waste is wet it rotates to 180 degree angle . And there is another flap like mechanism made by a servo motor which will open and close the bottom the flask in which the waste is put in and hence allowing us to control when and in which bin the waste would be put in .

2.4 Waste Level Detection

we have connected two ultrasonic distance sensors to a ESP module which when connected to a local wifi can wirelessly send the data to cloud database(we have used google's firebase)

2.5 Web App

We have built an web app on HTML,CSS, and Javascript and we have also used MIT app inventor to link this webapp to an android application. This webApp can retrieve data from the database and show the user lively the amount of waste level filled in each bin at any point in time.

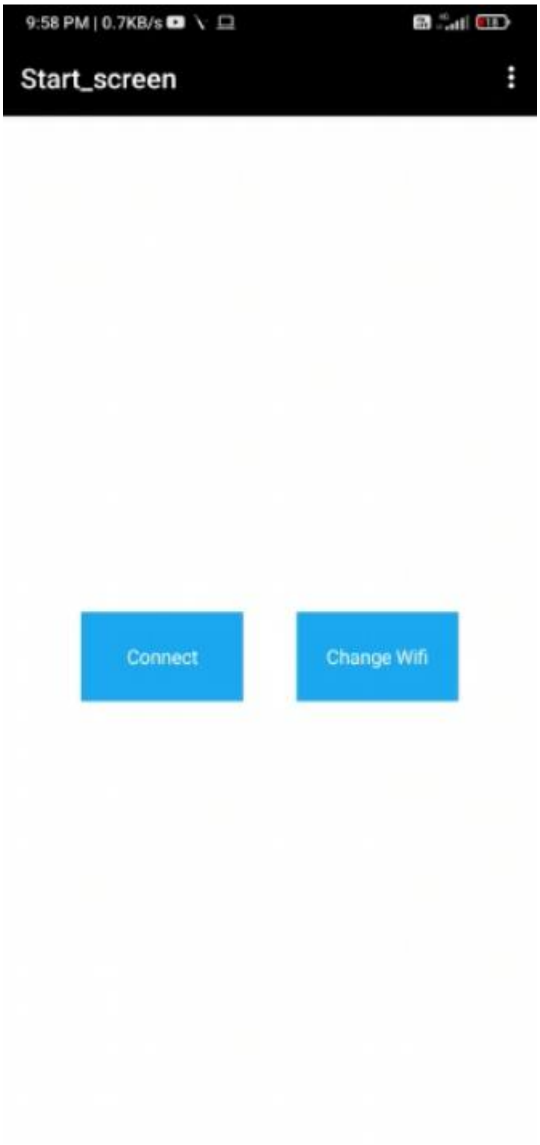
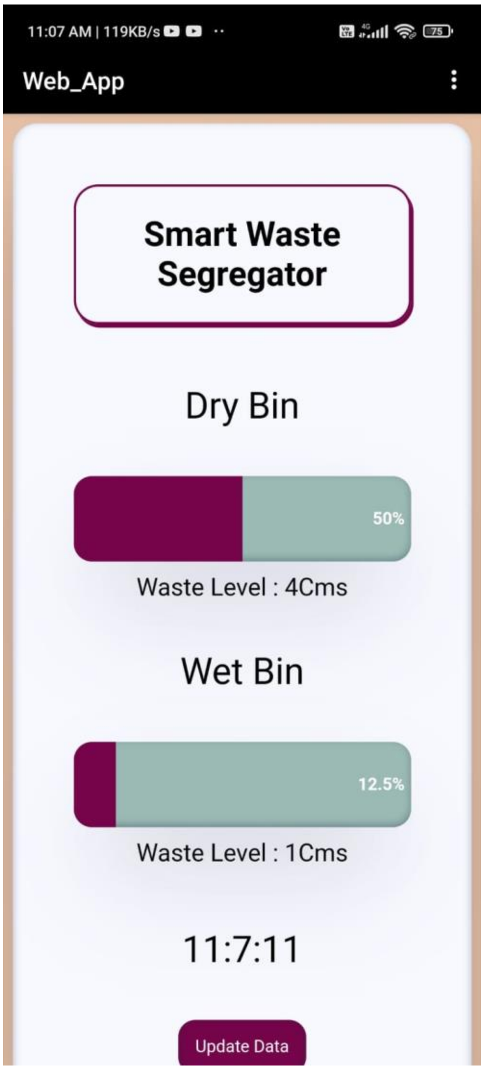


Figure 1 Home Screen of App



Smart Waste Segregator

Dry Bin

Waste Level : 4Cms

Wet Bin

Waste Level : 1Cms

11:7:11

Figure 2 User Interface of App

2.6 Working of the Device

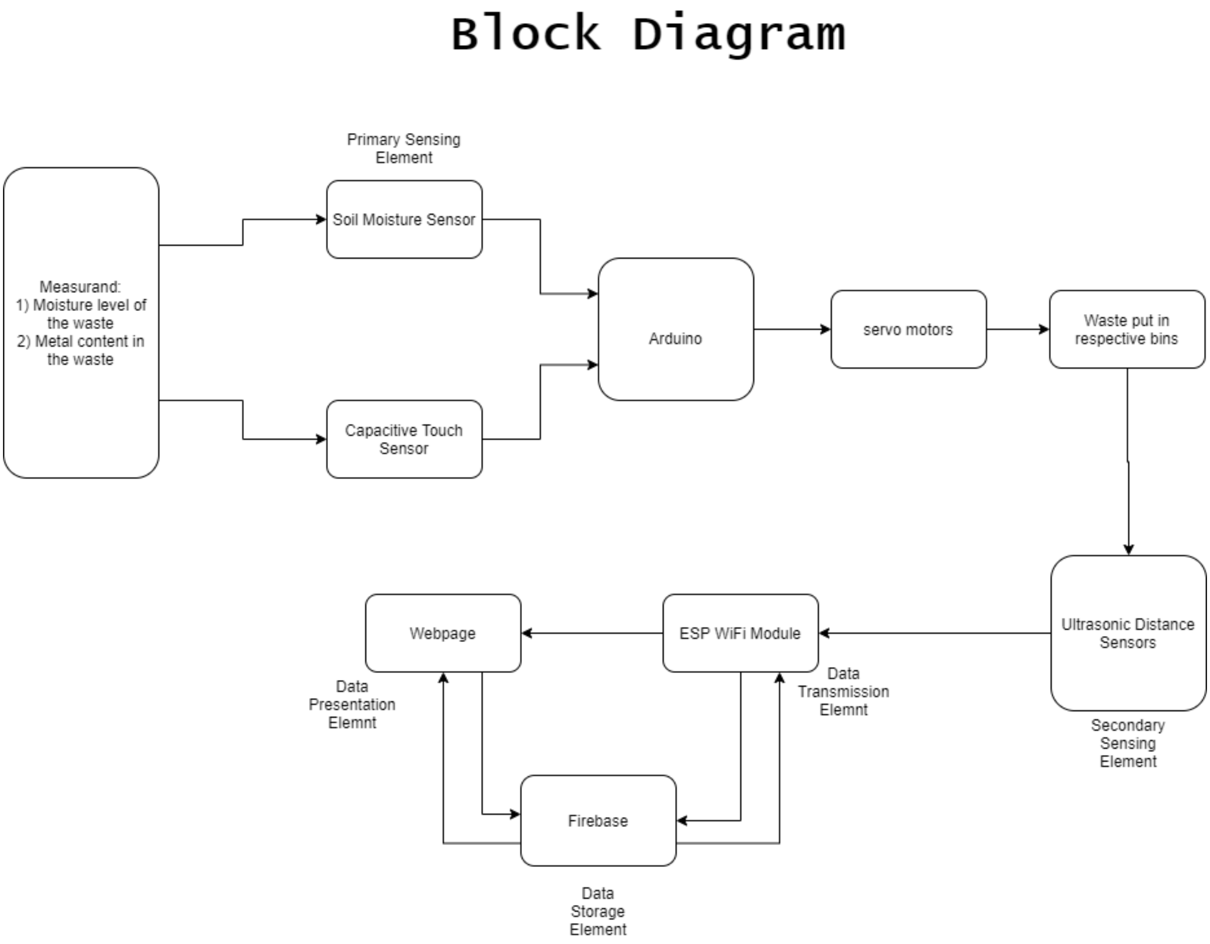


Figure 3 Functional Block Diagram of Device

Our device uses capacitive touch sensor to detect if there is any waste on it . If the waste is detected then it gets the moisture of the waste from the soil moisture and it then rotates the platform accordingly and then puts the waste in the right bin. The servo motors are controlled using Arduino And we have another circuit which is responsible for sending the waste levels to the server continuously to the firebase cloud server. Two ultrasonic sensors are connected to the ESP module which reads the level of garbage in each bin and updates the data in website every 4 seconds.

CHAPTER 3

Design of IoT based Waste Segregator

3.1 Design Approach

Our device has two primary functions . One is to segregate the waste the user puts in the correct bin automatically and other function of the device is to record the waste level in each bin and update in the cloud database which then can be used by our app to see the status of each bin at any point of time

We first drew a clear functional block diagram of the device. Then we proceeded to make Proteus simulation for the waste segregator . We were able to simulate the ultrasonic sensors, servo motors and touch sensors. But it was assuming several ideal conditions like we used a Boolean toggle to simulate to capcitive touch sensor which responds immediately to input but the actual sensor takes about 60ms to respond to the input. Once Protues simulation is done. We already had the baseline for the code for hardware. Then we created a web app using HTML,CSS, Javascript and we linked it to an App using MIT app inventor. We also designed the database during this time. Once that is done we made the physical components and completed the code for each of the ESP and Arduino Uno.

3.2 Realistic constrains

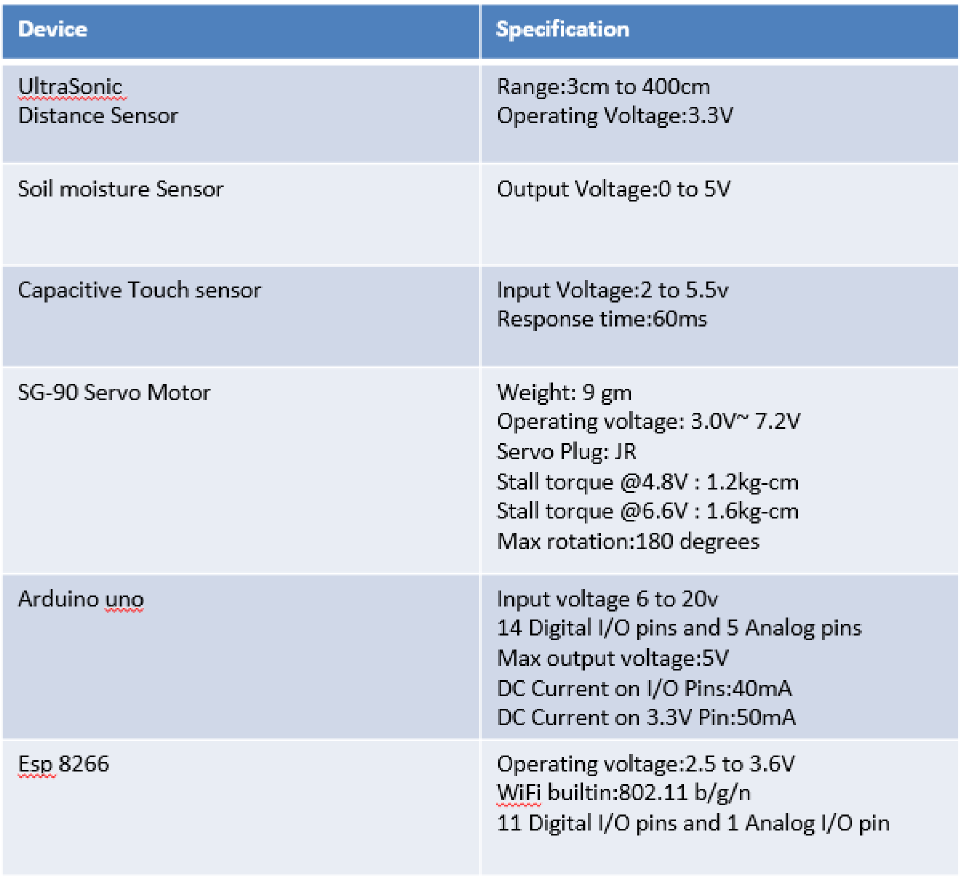
Our project is actually a mini model. In real we would life to use much larger motors and use a different mechanism to segregate the waste as moving a base with two large bins is really inefficient instead of just moving the funnel or opening in which the waste is being put in. Also in practical scenarios the Capacitive touch sensor might not respond to very light wastes like a thin paper which can reduce its functionality. Also we can build more complex mechanisms to dry out waste and remove moisture from it through other processes.

3.3 Alternatives and Tradeoffs

We could have used strain gauge or other force detecting sensor also to see if the device needs to be turned on. Though segregating the waste into dry and wet makes the disposal more efficient it could be further improved by also segregating the waste into more categories such as metal waste, plastic waste which can be implemented in future.

3.4 Design Specifications

Table 1 Design Specification



Device	Specification
UltraSonic Distance Sensor	Range:3cm to 400cm Operating Voltage:3.3V
Soil moisture Sensor	Output Voltage:0 to 5V
Capacitive Touch sensor	Input Voltage:2 to 5.5v Response time:60ms
SG-90 Servo Motor	Weight: 9 gm Operating voltage: 3.0V~ 7.2V Servo Plug: JR Stall torque @4.8V : 1.2kg-cm Stall torque @6.6V : 1.6kg-cm Max rotation:180 degrees
Arduino uno	Input voltage 6 to 20v 14 Digital I/O pins and 5 Analog pins Max output voltage:5V DC Current on I/O Pins:40mA DC Current on 3.3V Pin:50mA
Esp 8266	Operating voltage:2.5 to 3.6V WiFi builtin:802.11 b/g/n 11 Digital I/O pins and 1 Analog I/O pin

The range of the Ultrasonic sensor is 3cm to 400cm which is a wide range considering the cost of the sensor and it can easily be connected to Arduino or Esp Module to get the distance. The input range of the capacitive senor is from 2 to 5.5v so we have directly connected its Vcc to 5v power from Arduino Uno. The response time of the Capacitive touch sensor is 60ms so the minimum time interval for which the device can check for the presence of waste is 60ms. The servo motors can around 1.4kg-cm torque so it capable of rotating the bins and the platform. We have connected the ESP to firebase database the time taken for communication is around 3 to 4 seconds so our device can only send data only once in 4 seconds as connecting to firebase cloud takes so

CHAPTER 4

PROJECT DEMONSTRATION

4.1 Introduction

Our project has two primary functions

- *Automate Segregation of the waste into three bins i.e. dry and wet waste. We have achieved this with the use of soil moisture sensor which senses the wetness of the waste and digital capactive senor which senses the presence of waste and two servo motors which aids us in making the mechanism to put the waste in correct bin according its characteristiscs all these sensors are controlled by Arduino Uno*
- *Indicate the Filled Capacity in each bin using IOT and APP.*
  We have used Esp8266 IoT module and ultrasonic distance sensors to achieve this purpose. The ultrasonic distance sensors give the amount of space filled in each bin which is then uploaded to a cloud database known as firebase with the help of ESP module. The data from the cloud database can be accessed through our Custom made Mobile App.

4.2 Analytic Results

Distance Calculation from inputs from ultra sonic distance sensor

Trig pin sends a pulse of frequency of 40khz and then once it hits and returns back turns echo pin high. We can get the time from Pulsein function()

*Distance = Vspeedofsound * time;*

Moisture level from Soil moisture sensor

Max Voltage out = 5V

No of values = 1024

*Moisture level% = AnalogRead/1024 *100*

4.3 Simulation Results

Since we cannot simulate ESP module using Protues we have used Arduino to showcase all the features of the device.

Simulation Circuit

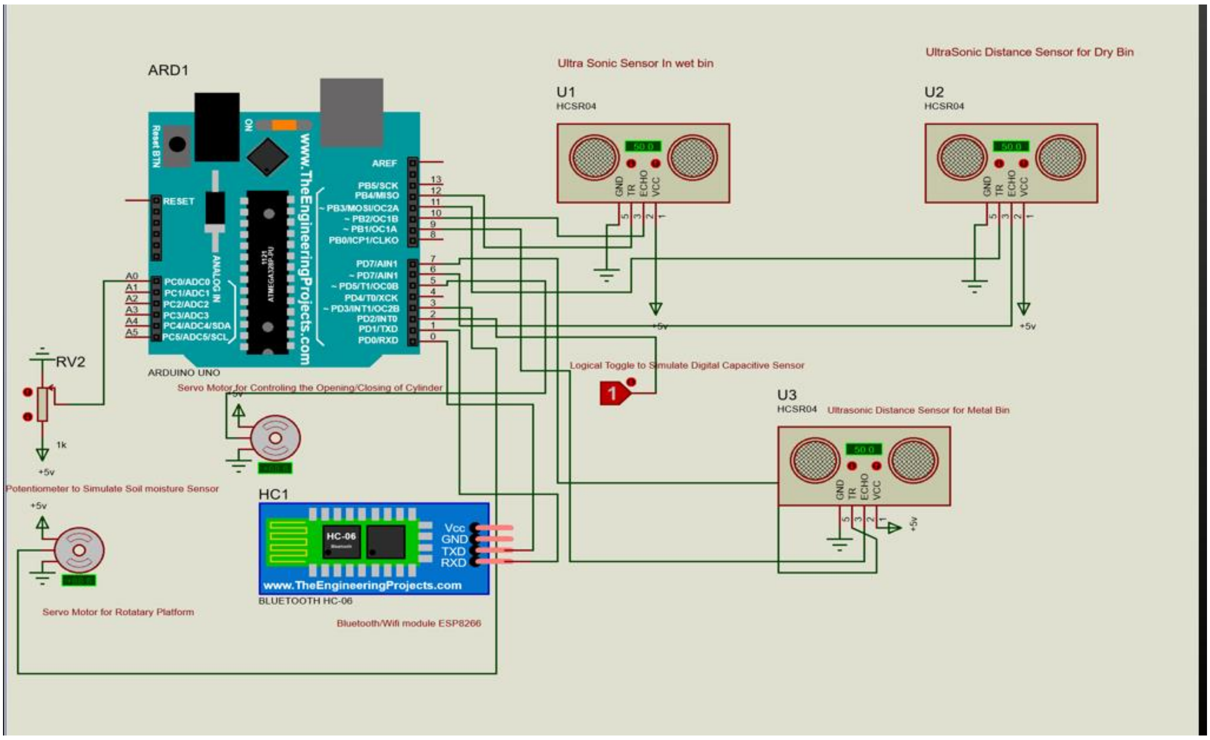
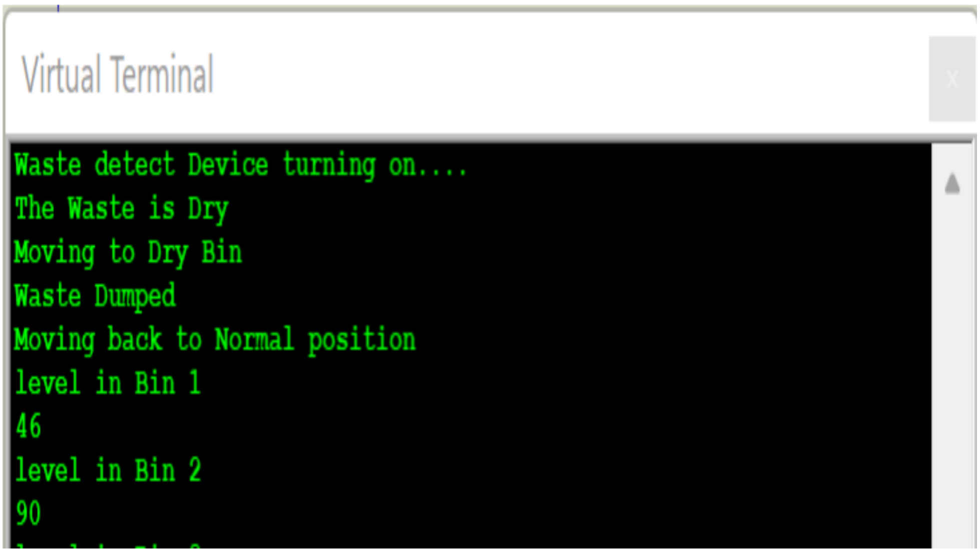


Figure 4 Simulation Circuit

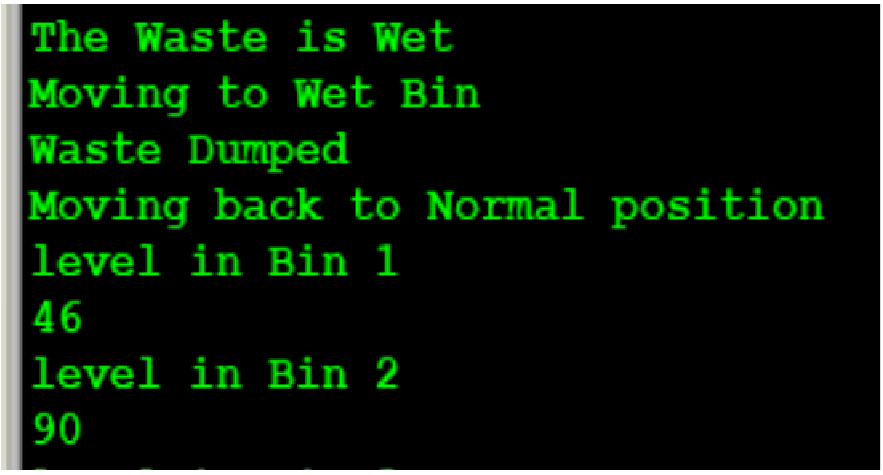
Case 1 when the waste is dry



```
Waste detect Device turning on....
The Waste is Dry
Moving to Dry Bin
Waste Dumped
Moving back to Normal position
level in Bin 1
46
level in Bin 2
90
```

Figure 5 Simulation Output Case 2

Case 2 when the waste is wet

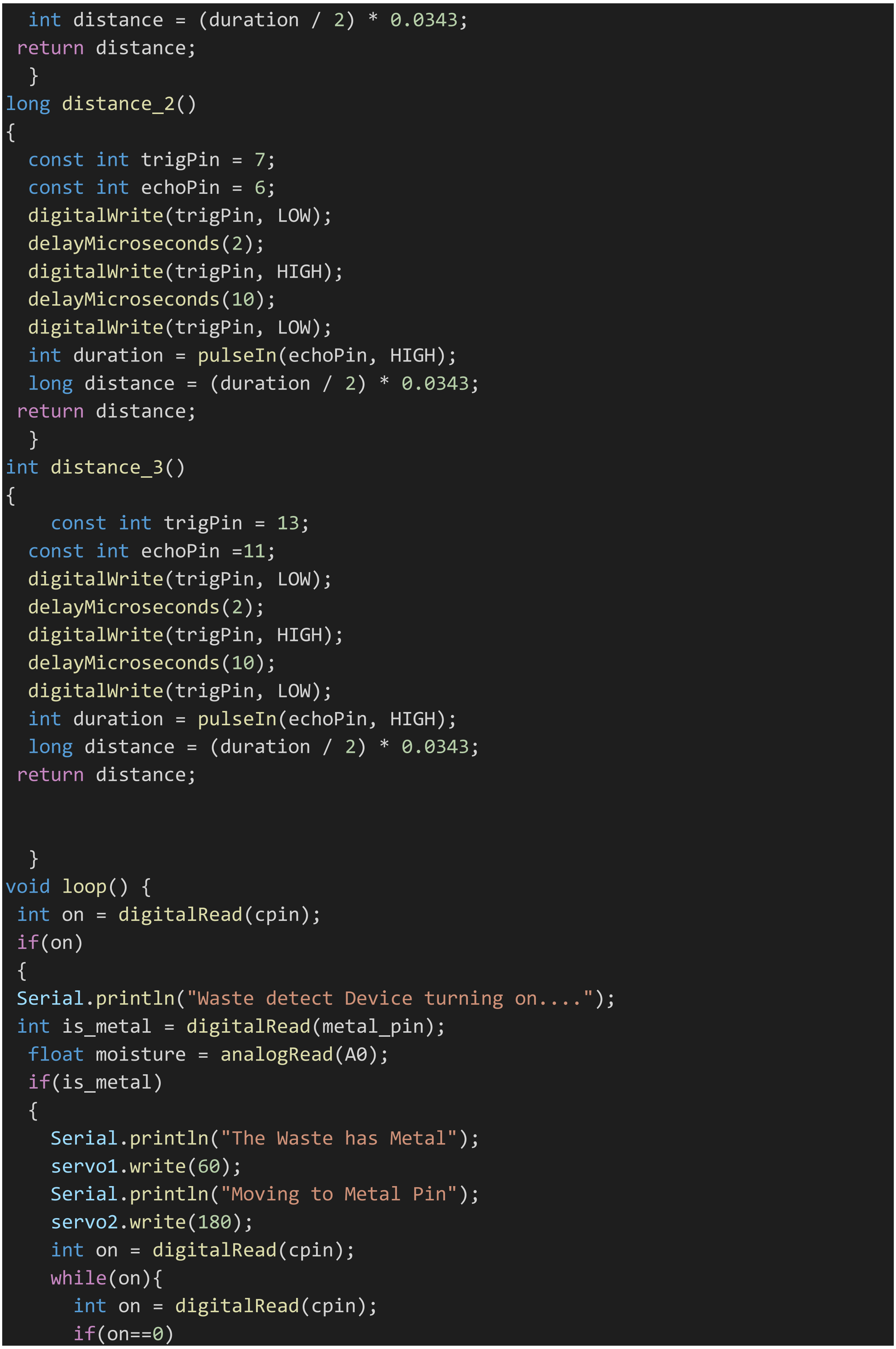


```
The Waste is Wet
Moving to Wet Bin
Waste Dumped
Moving back to Normal position
level in Bin 1
46
level in Bin 2
90
```

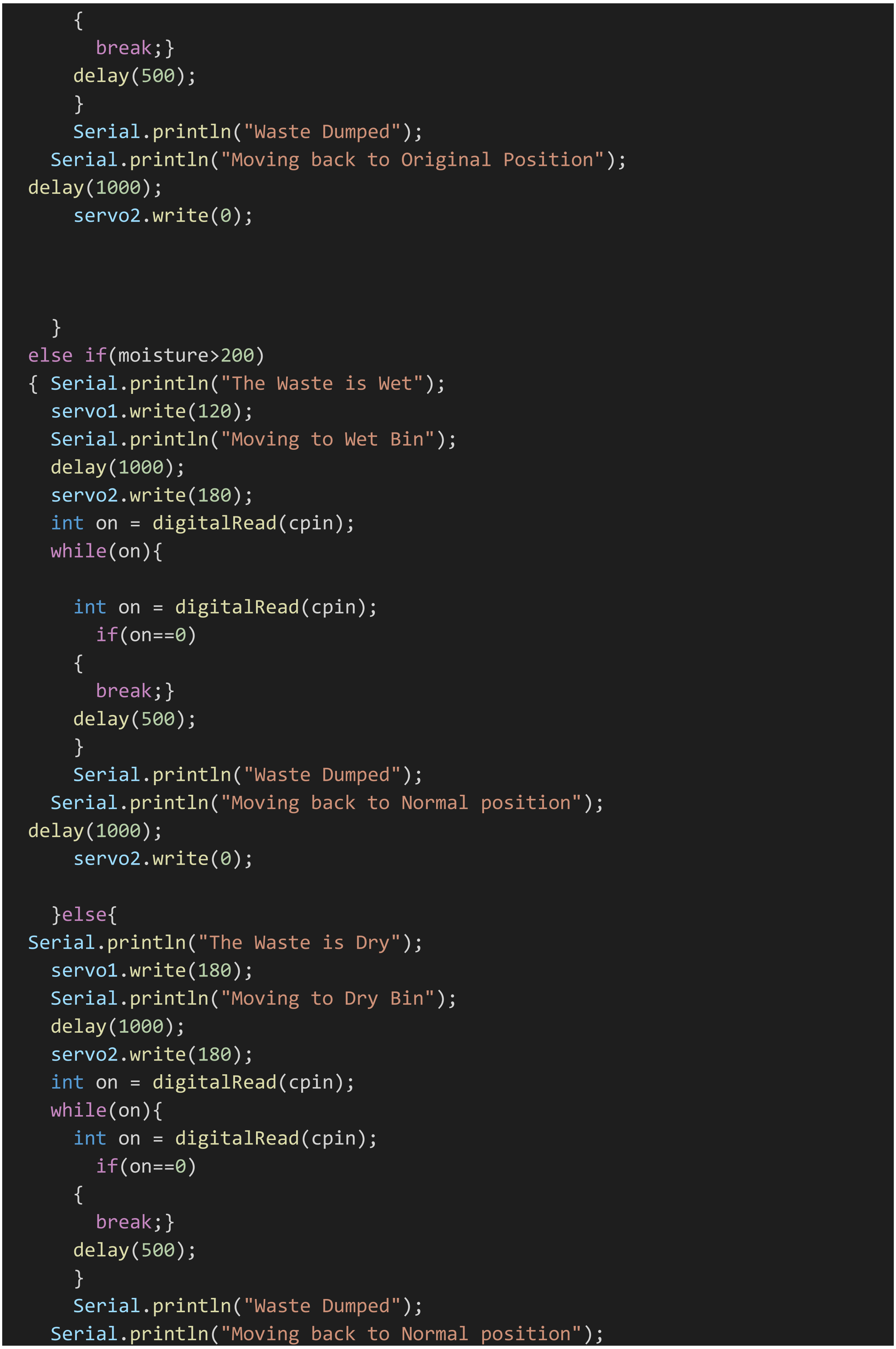
Figure 6 Simulation Output Case 2

Code for Simulation

```
#include<Servo.h>

int servopin1 =5 ;
int servopin2 = 3;
const int cpin = 2;
const int metal_pin = 12;
Servo servo1;
Servo servo2;
void setup() {
   Serial.begin(9600);// Starting Serial Terminal
   pinMode(cpin,INPUT);
   servo1.attach(servopin1);
   servo2.attach(servopin2);
}


int distance_1()
{
  const int trigPin = 9;
  const int echoPin = 10;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int duration = pulseIn(echoPin, HIGH);
```



```
  int distance = (duration / 2) * 0.0343;
 return distance;
  }
long distance_2()
{
  const int trigPin = 7;
  const int echoPin = 6;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int duration = pulseIn(echoPin, HIGH);
  long distance = (duration / 2) * 0.0343;
 return distance;
  }
int distance_3()
{
    const int trigPin = 13;
  const int echoPin =11;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int duration = pulseIn(echoPin, HIGH);
  long distance = (duration / 2) * 0.0343;
 return distance;


  }
void loop() {
 int on = digitalRead(cpin);
 if(on)
 {
 Serial.println("Waste detect Device turning on....");
 int is_metal = digitalRead(metal_pin);
  float moisture = analogRead(A0);
  if(is_metal)
  {
    Serial.println("The Waste has Metal");
    servo1.write(60);
    Serial.println("Moving to Metal Pin");
    servo2.write(180);
    int on = digitalRead(cpin);
    while(on){
      int on = digitalRead(cpin);
      if(on==0)
```



```
  {
   break;}
  delay(500);
  }
  Serial.println("Waste Dumped");
 Serial.println("Moving back to Original Position");
delay(1000);
  servo2.write(0);



 }
else if(moisture>200)
{ Serial.println("The Waste is Wet");
 servo1.write(120);
 Serial.println("Moving to Wet Bin");
 delay(1000);
 servo2.write(180);
 int on = digitalRead(cpin);
 while(on){

  int on = digitalRead(cpin);
   if(on==0)
  {
   break;}
  delay(500);
  }
  Serial.println("Waste Dumped");
 Serial.println("Moving back to Normal position");
delay(1000);
  servo2.write(0);

 }else{
Serial.println("The Waste is Dry");
 servo1.write(180);
 Serial.println("Moving to Dry Bin");
 delay(1000);
 servo2.write(180);
 int on = digitalRead(cpin);
 while(on){
  int on = digitalRead(cpin);
   if(on==0)
  {
   break;}
  delay(500);
  }
  Serial.println("Waste Dumped");
 Serial.println("Moving back to Normal position");
```

```
delay(1000);
    servo2.write(0);


    }
}
```

4.4 Hardware Results

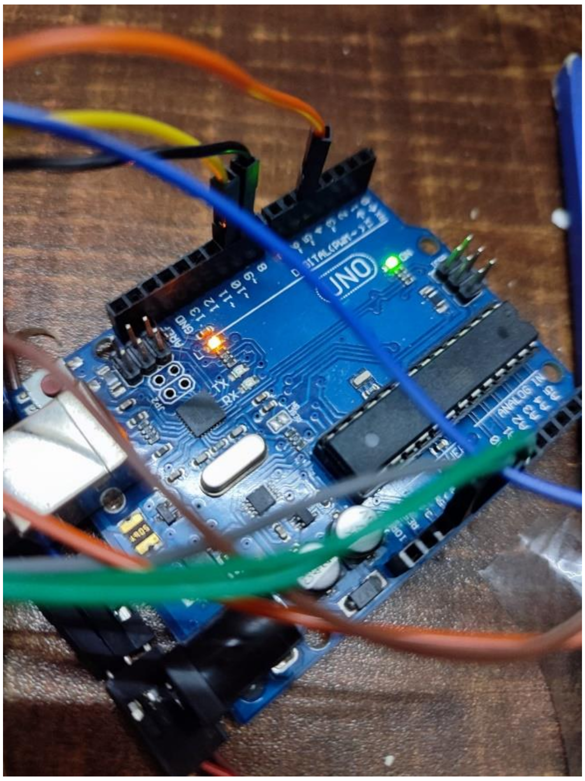


Figure 7 Arduino Uno

Arduino Uno

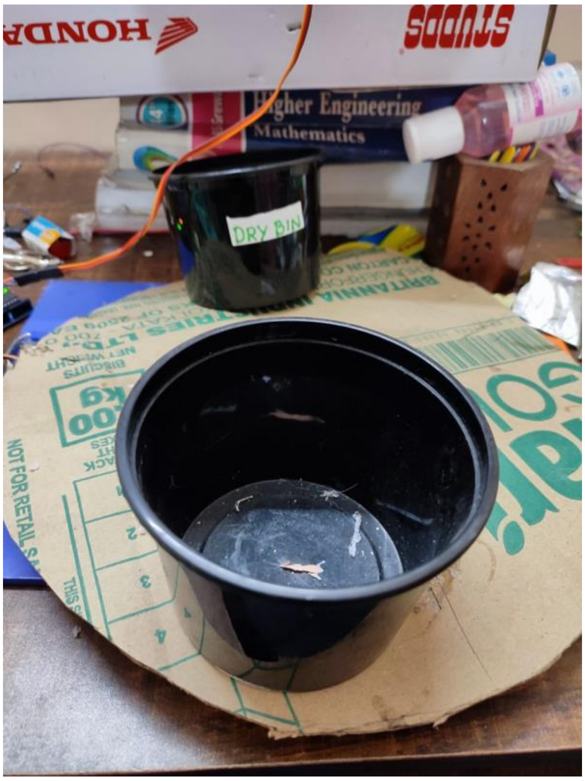


Figure 8 Bins and Rotating Platform

Wet and Dry Bins

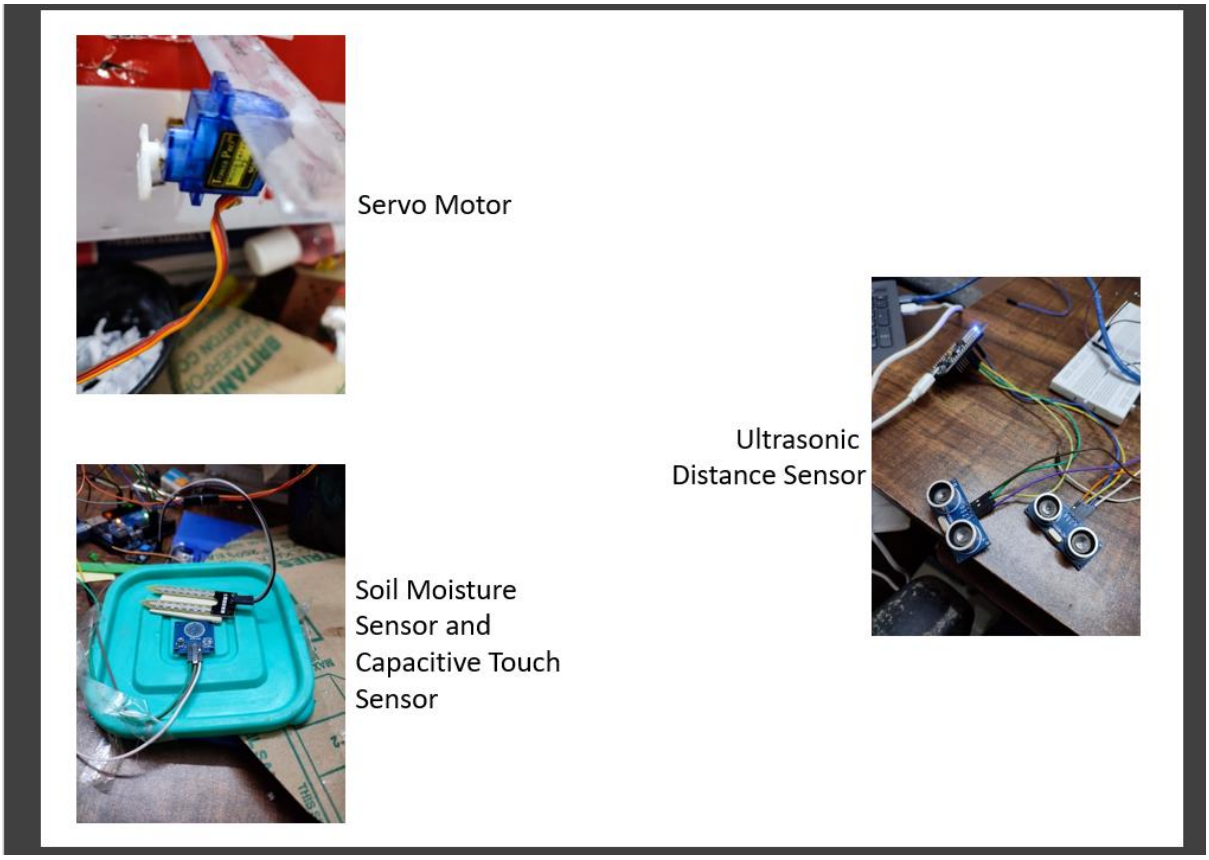

Servo Motor
Ultrasonic Distance Sensor
Soil Moisture Sensor and Capacitive Touch Sensor

Figure 9 Sensors and Servo Motors

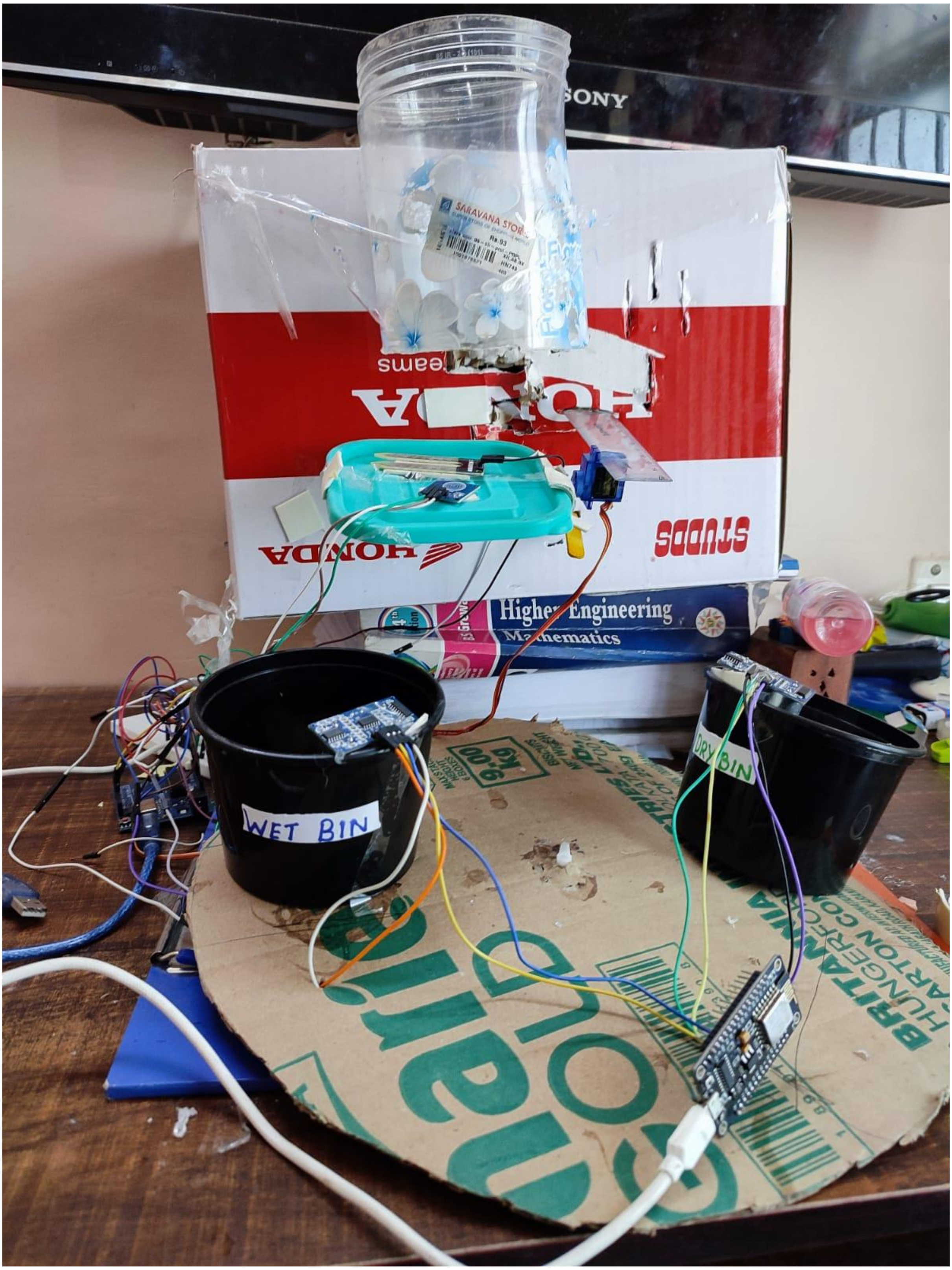
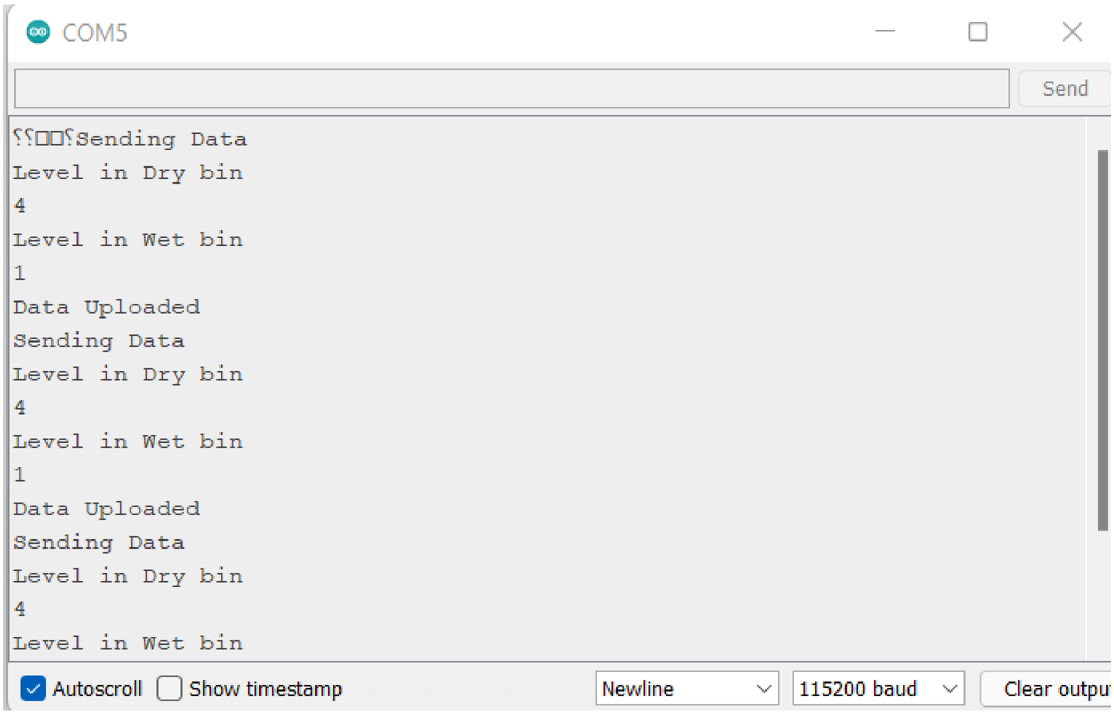


Figure 10 Image of Final Hardware

Output From Esp Module



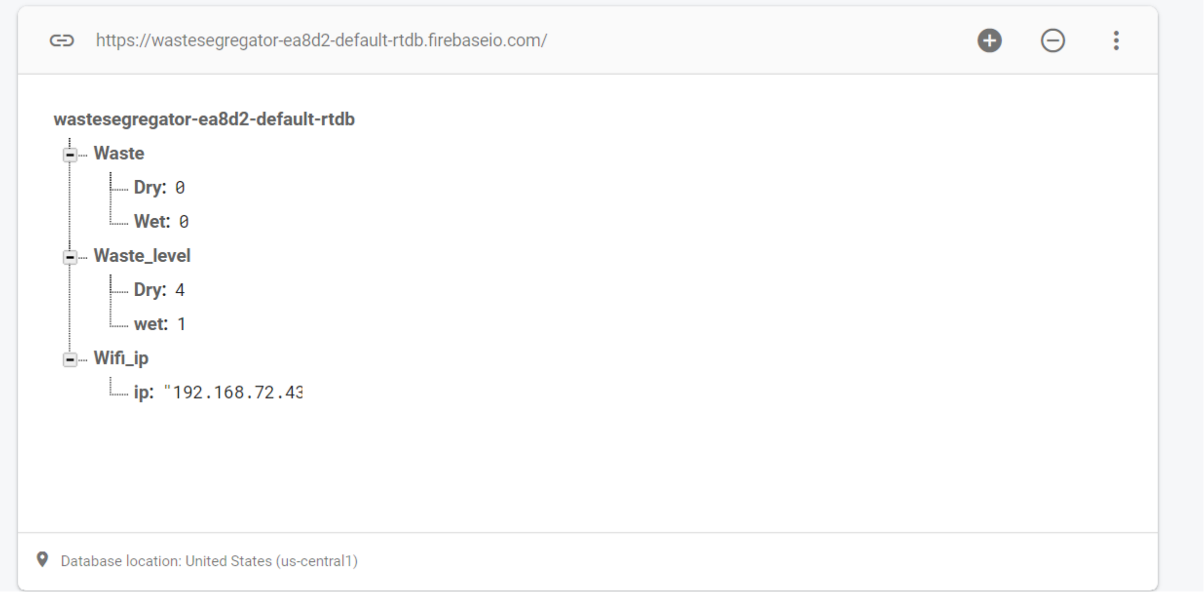
COM5

```
⸮⸮□□⸮Sending Data
Level in Dry bin
4
Level in Wet bin
1
Data Uploaded
Sending Data
Level in Dry bin
4
Level in Wet bin
1
Data Uploaded
Sending Data
Level in Dry bin
4
Level in Wet bin
```

☑ Autoscroll □ Show timestamp Newline 115200 baud

Figure 11 Serial Monitor Output of ESP8266

Data Stored in Firebase



https://wastesegregator-ea8d2-default-rtdb.firebaseio.com/

```
wastesegregator-ea8d2-default-rtdb
  Waste
    Dry: 0
    Wet: 0
  Waste_level
    Dry: 4
    wet: 1
  Wifi_ip
    ip: "192.168.72.43
```

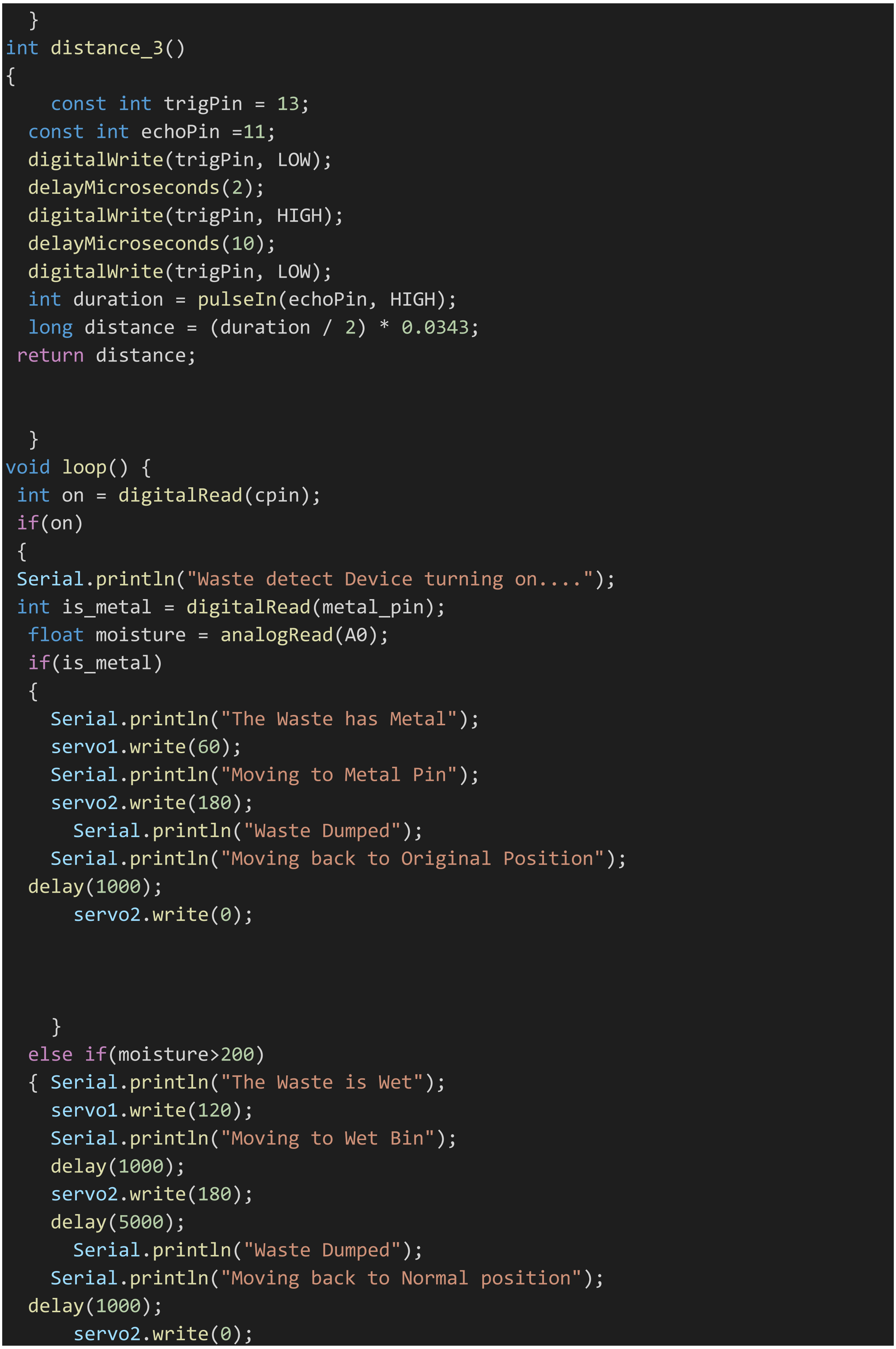
Database location: United States (us-central1)

Figure 12 Data Stored in Firebase

Code for Actual Hardware

Arduino

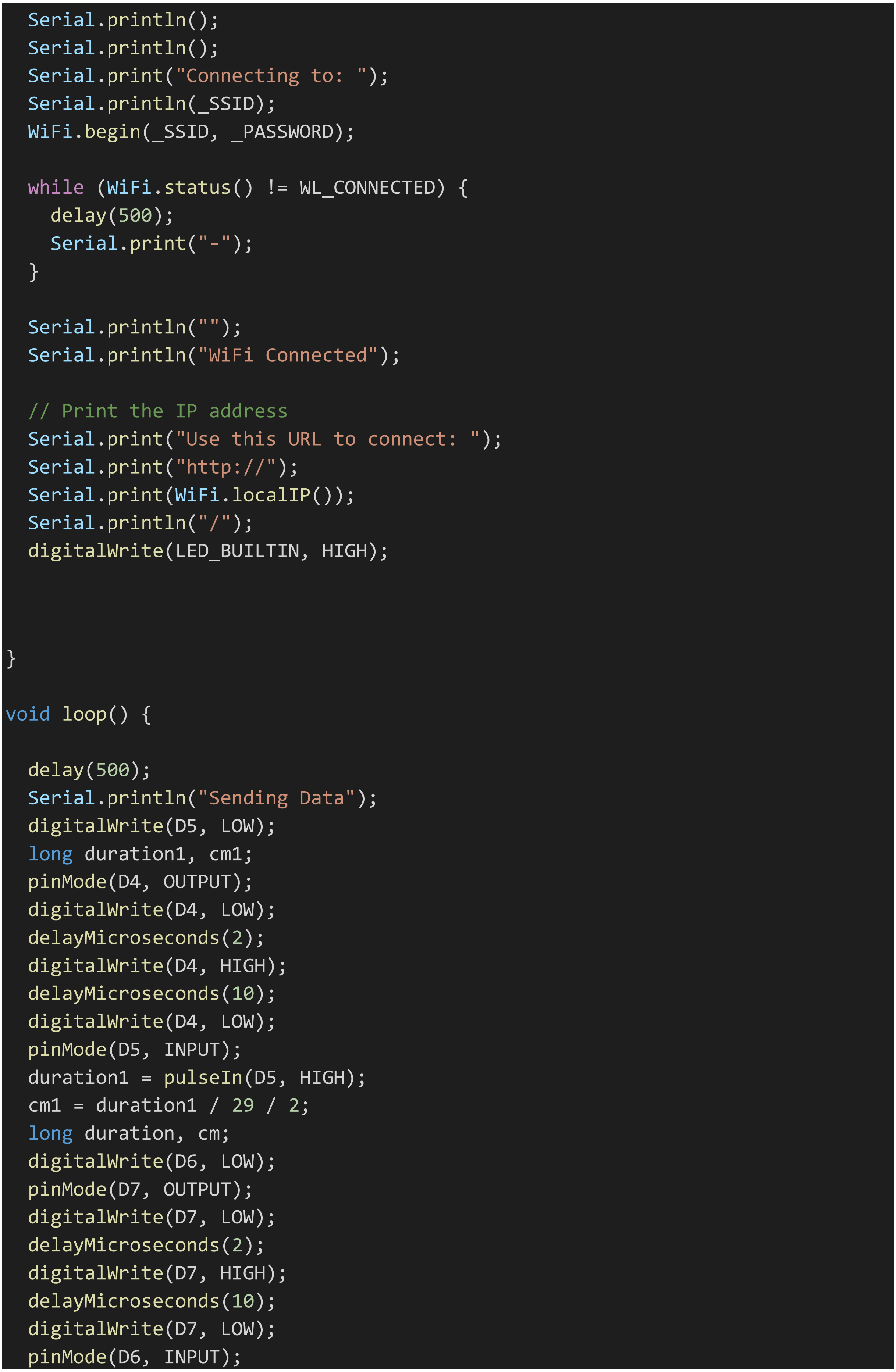
```
#include<Servo.h>

int servopin1 =5 ;
int servopin2 = 3;
const int cpin = 2;
const int metal_pin = 12;
Servo servo1;
Servo servo2;
void setup() {
   Serial.begin(9600);// Starting Serial Terminal
   pinMode(cpin,INPUT);
   servo1.attach(servopin1);
   servo2.attach(servopin2);
}


int distance_1()
{
  const int trigPin = 9;
  const int echoPin = 10;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int duration = pulseIn(echoPin, HIGH);
  int distance = (duration / 2) * 0.0343;
 return distance;
  }
long distance_2()
{
  const int trigPin = 7;
  const int echoPin = 6;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int duration = pulseIn(echoPin, HIGH);
  long distance = (duration / 2) * 0.0343;
 return distance;
```



```
  }
int distance_3()
{
    const int trigPin = 13;
  const int echoPin =11;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int duration = pulseIn(echoPin, HIGH);
  long distance = (duration / 2) * 0.0343;
 return distance;


  }
void loop() {
 int on = digitalRead(cpin);
 if(on)
 {
 Serial.println("Waste detect Device turning on....");
 int is_metal = digitalRead(metal_pin);
  float moisture = analogRead(A0);
  if(is_metal)
  {
    Serial.println("The Waste has Metal");
    servo1.write(60);
    Serial.println("Moving to Metal Pin");
    servo2.write(180);
      Serial.println("Waste Dumped");
    Serial.println("Moving back to Original Position");
  delay(1000);
      servo2.write(0);



    }
  else if(moisture>200)
  { Serial.println("The Waste is Wet");
    servo1.write(120);
    Serial.println("Moving to Wet Bin");
    delay(1000);
    servo2.write(180);
    delay(5000);
      Serial.println("Waste Dumped");
    Serial.println("Moving back to Normal position");
  delay(1000);
      servo2.write(0);
  }else{
 Serial.println("The Waste is Dry");
  servo1.write(180);
  Serial.println("Moving to Dry Bin");
  delay(5000);
  servo2.write(180);

   Serial.println("Waste Dumped");
  Serial.println("Moving back to Normal position");
 delay(1000);
   servo2.write(0);


   }
 }
```

ESP8266 Module

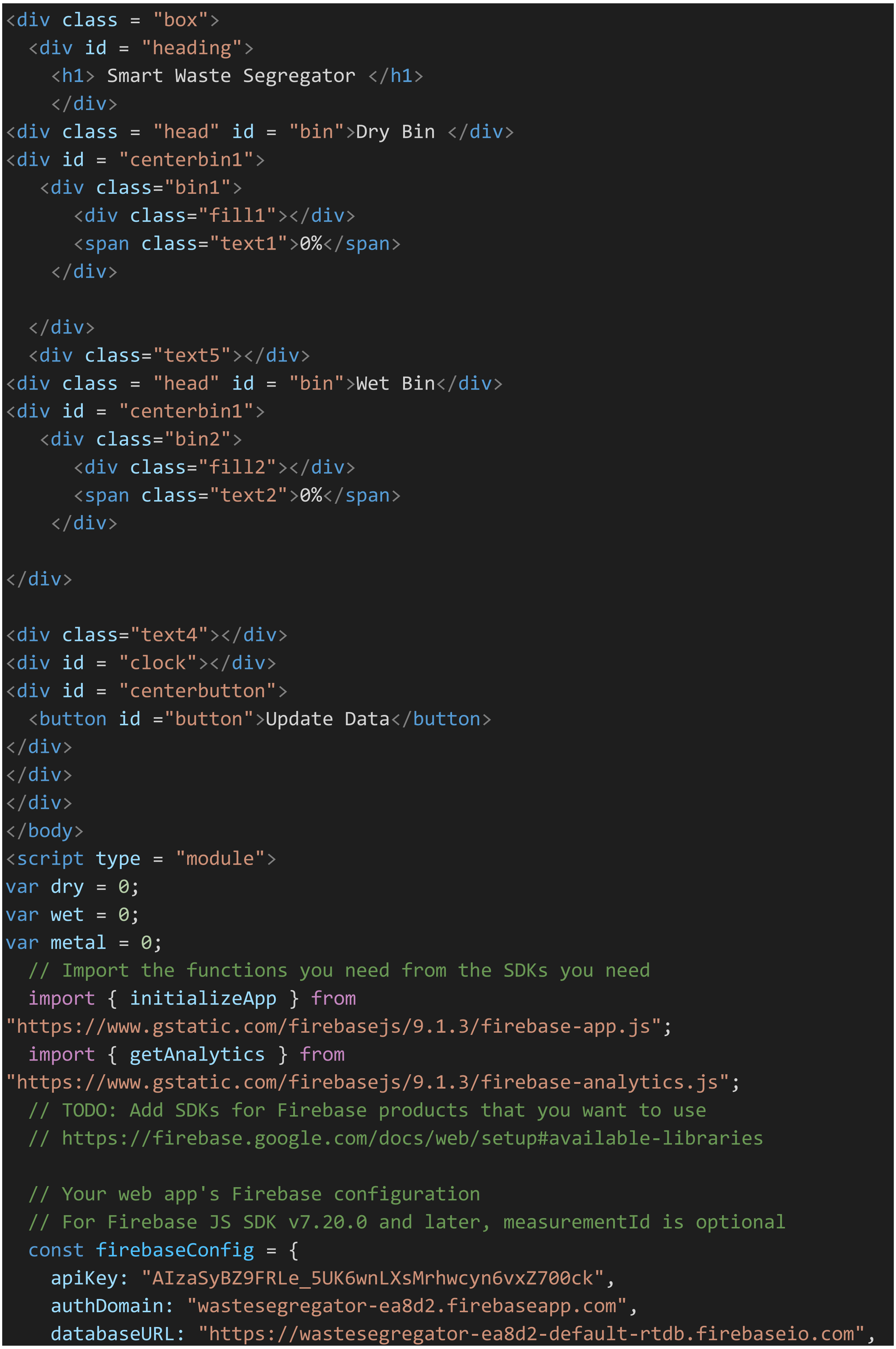
```
#include <ESP8266Firebase.h>
#include <ESP8266WiFi.h>

#define _SSID "Redmix"        // Your WiFi SSID
#define _PASSWORD "redmi1021"    // Your WiFi Password  // Your Firebase Project ID. Can be found in project settings.
#define PROJECT_ID "wastesegregator-ea8d2-default-rtdb"
const int cpin = 5;
Firebase firebase(PROJECT_ID);



void setup() {
  pinMode(D6, INPUT);
  pinMode(D7, OUTPUT);
  pinMode(D7, INPUT);
  pinMode(D4, OUTPUT);
  Serial.begin(115200);
  Serial.println("test");
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(1000);
```



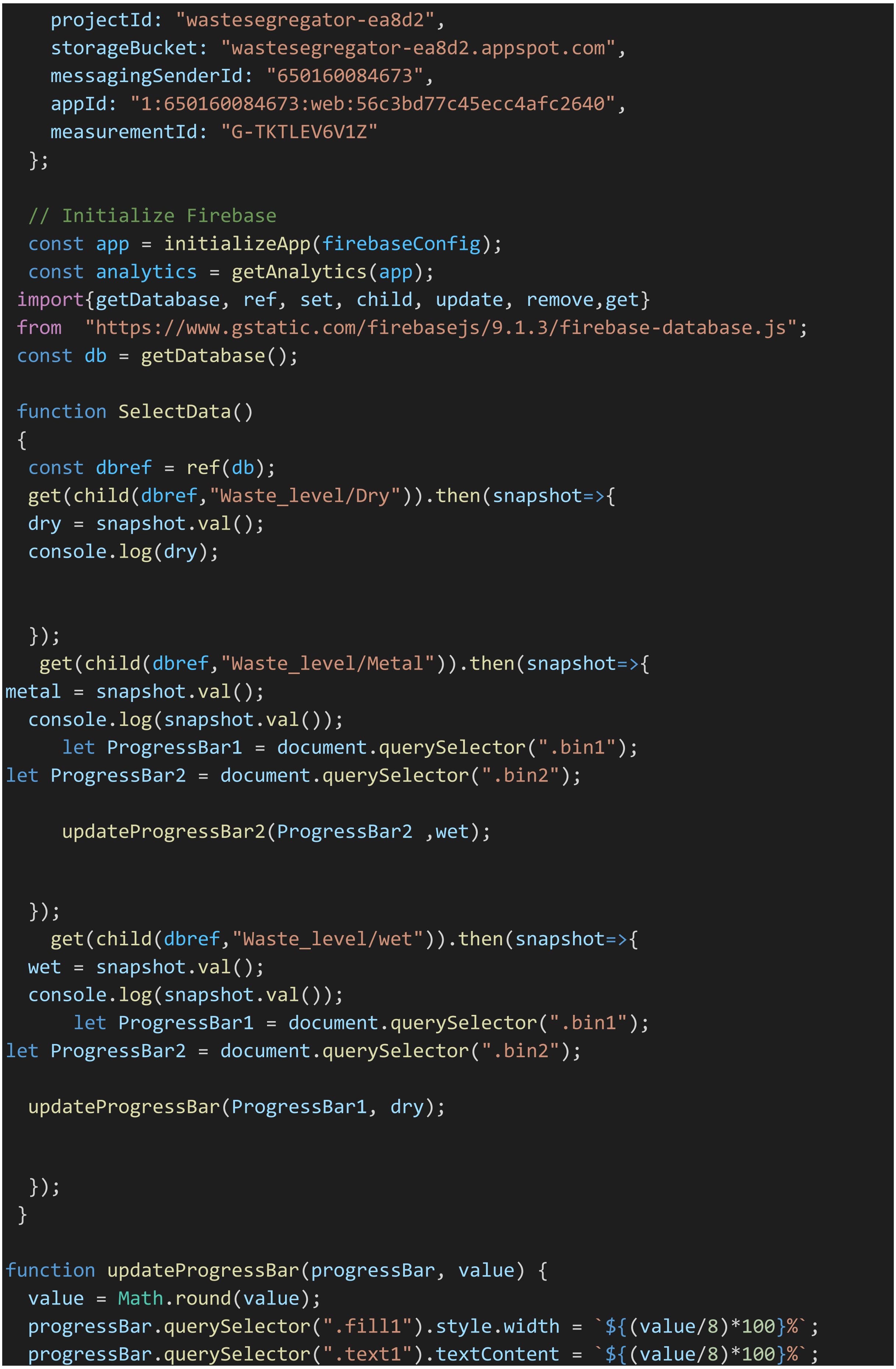
```
  Serial.println();
  Serial.println();
  Serial.print("Connecting to: ");
  Serial.println(_SSID);
  WiFi.begin(_SSID, _PASSWORD);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print("-");
  }

  Serial.println("");
  Serial.println("WiFi Connected");

  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
  digitalWrite(LED_BUILTIN, HIGH);



}

void loop() {

  delay(500);
  Serial.println("Sending Data");
  digitalWrite(D5, LOW);
  long duration1, cm1;
  pinMode(D4, OUTPUT);
  digitalWrite(D4, LOW);
  delayMicroseconds(2);
  digitalWrite(D4, HIGH);
  delayMicroseconds(10);
  digitalWrite(D4, LOW);
  pinMode(D5, INPUT);
  duration1 = pulseIn(D5, HIGH);
  cm1 = duration1 / 29 / 2;
  long duration, cm;
  digitalWrite(D6, LOW);
  pinMode(D7, OUTPUT);
  digitalWrite(D7, LOW);
  delayMicroseconds(2);
  digitalWrite(D7, HIGH);
  delayMicroseconds(10);
  digitalWrite(D7, LOW);
  pinMode(D6, INPUT);
  duration = pulseIn(D6, HIGH);
  cm = duration / 29 / 2;

  firebase.setFloat("Waste_level/Dry", cm);
  firebase.setFloat("Waste_level/wet", cm1);
  Serial.println("Level in Dry bin");
  Serial.println(8-cm);
  Serial.println("Level in Wet bin");
  Serial.println(8-cm1);
  Serial.println("Data Uploaded");
}
```

WebAPP

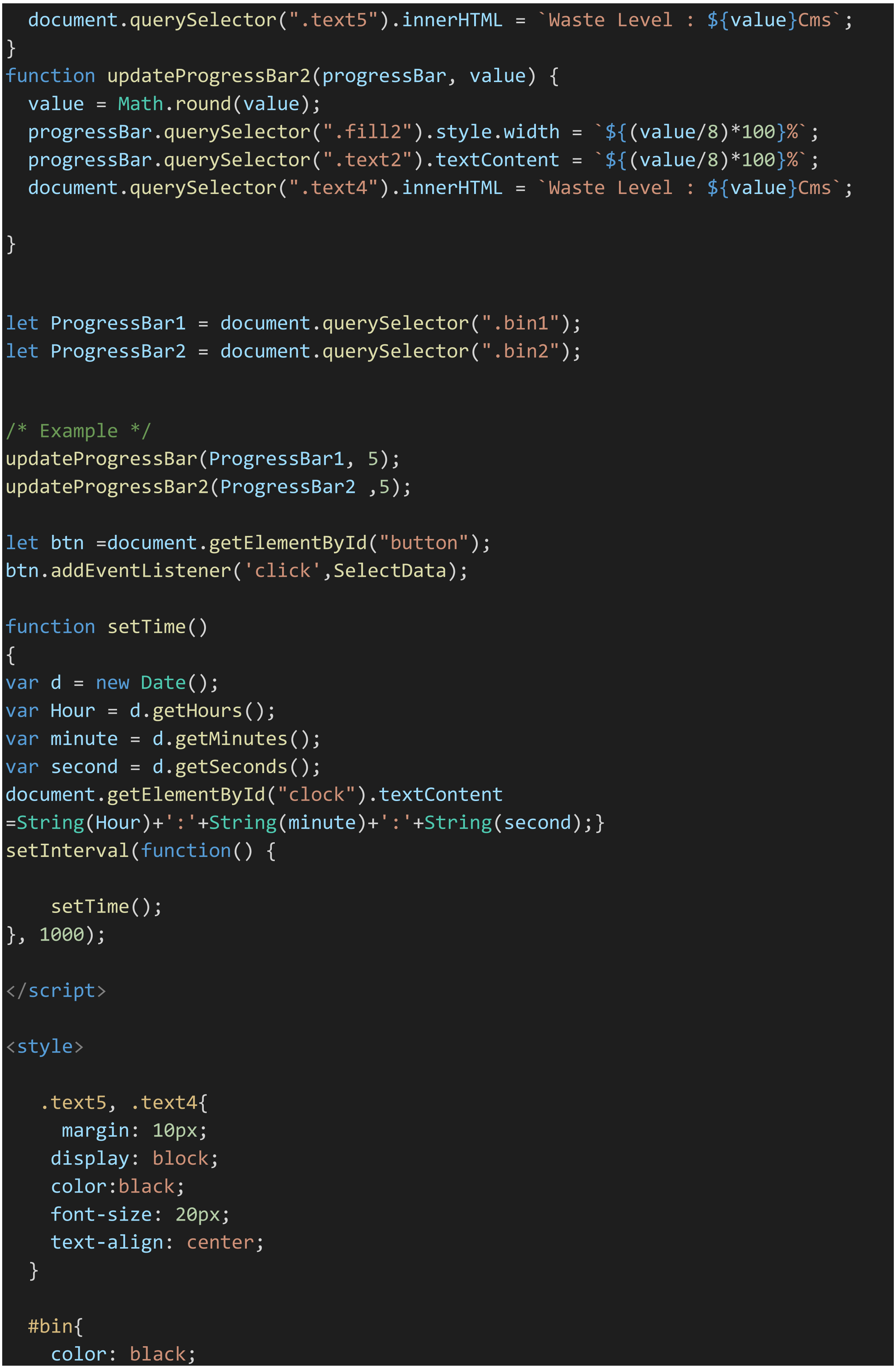
```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scrip
t>



    <title>Smart Waste Segregator</title>

  </head>
<body>
```



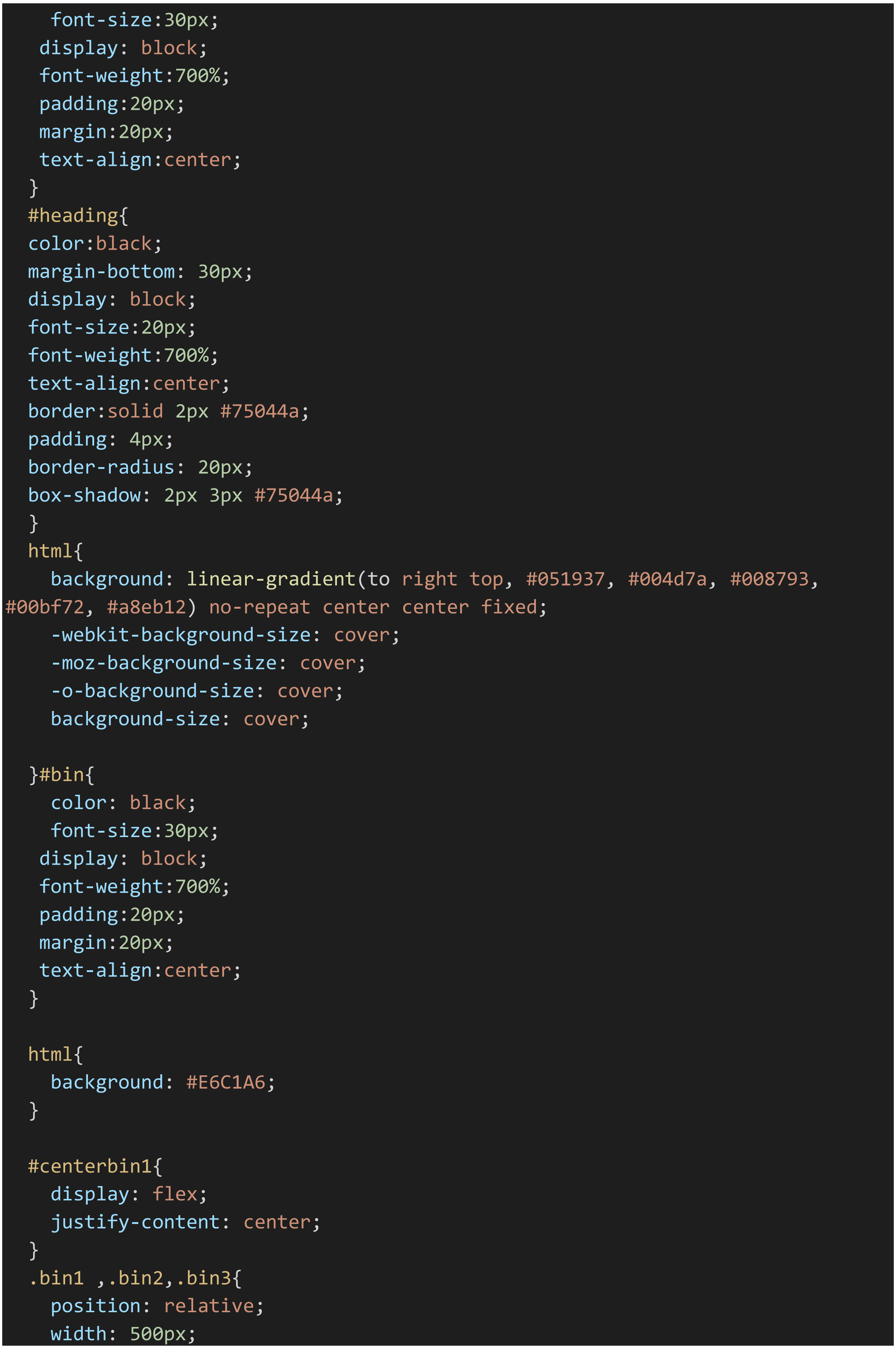
```
<div class = "box">
  <div id = "heading">
    <h1> Smart Waste Segregator </h1>
    </div>
<div class = "head" id = "bin">Dry Bin </div>
<div id = "centerbin1">
   <div class="bin1">
      <div class="fill1"></div>
      <span class="text1">0%</span>
    </div>

  </div>
  <div class="text5"></div>
<div class = "head" id = "bin">Wet Bin</div>
<div id = "centerbin1">
   <div class="bin2">
      <div class="fill2"></div>
      <span class="text2">0%</span>
    </div>

</div>

<div class="text4"></div>
<div id = "clock"></div>
<div id = "centerbutton">
  <button id ="button">Update Data</button>
</div>
</div>
</div>
</body>
<script type = "module">
var dry = 0;
var wet = 0;
var metal = 0;
  // Import the functions you need from the SDKs you need
  import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.1.3/firebase-app.js";
  import { getAnalytics } from
"https://www.gstatic.com/firebasejs/9.1.3/firebase-analytics.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  const firebaseConfig = {
    apiKey: "AIzaSyBZ9FRLe_5UK6wnLXsMrhwcyn6vxZ700ck",
    authDomain: "wastesegregator-ea8d2.firebaseapp.com",
    databaseURL: "https://wastesegregator-ea8d2-default-rtdb.firebaseio.com",
```



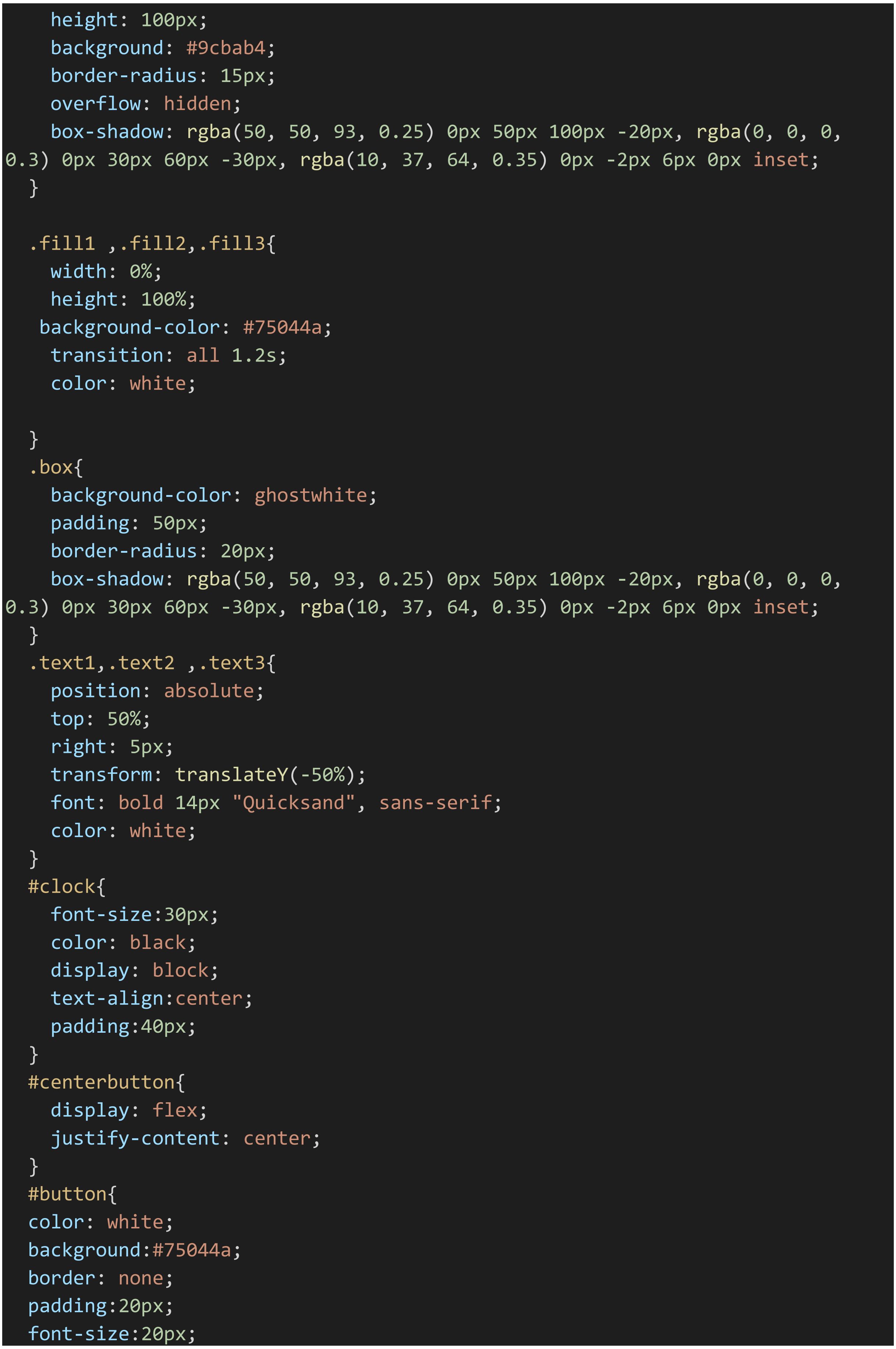
```
    projectId: "wastesegregator-ea8d2",
    storageBucket: "wastesegregator-ea8d2.appspot.com",
    messagingSenderId: "650160084673",
    appId: "1:650160084673:web:56c3bd77c45ecc4afc2640",
    measurementId: "G-TKTLEV6V1Z"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
 import{getDatabase, ref, set, child, update, remove,get}
 from  "https://www.gstatic.com/firebasejs/9.1.3/firebase-database.js";
 const db = getDatabase();

 function SelectData()
 {
  const dbref = ref(db);
  get(child(dbref,"Waste_level/Dry")).then(snapshot=>{
  dry = snapshot.val();
  console.log(dry);


  });
   get(child(dbref,"Waste_level/Metal")).then(snapshot=>{
metal = snapshot.val();
  console.log(snapshot.val());
     let ProgressBar1 = document.querySelector(".bin1");
let ProgressBar2 = document.querySelector(".bin2");

     updateProgressBar2(ProgressBar2 ,wet);


  });
    get(child(dbref,"Waste_level/wet")).then(snapshot=>{
  wet = snapshot.val();
  console.log(snapshot.val());
      let ProgressBar1 = document.querySelector(".bin1");
let ProgressBar2 = document.querySelector(".bin2");

  updateProgressBar(ProgressBar1, dry);


  });
 }

function updateProgressBar(progressBar, value) {
  value = Math.round(value);
  progressBar.querySelector(".fill1").style.width = `${(value/8)*100}%`;
  progressBar.querySelector(".text1").textContent = `${(value/8)*100}%`;
```



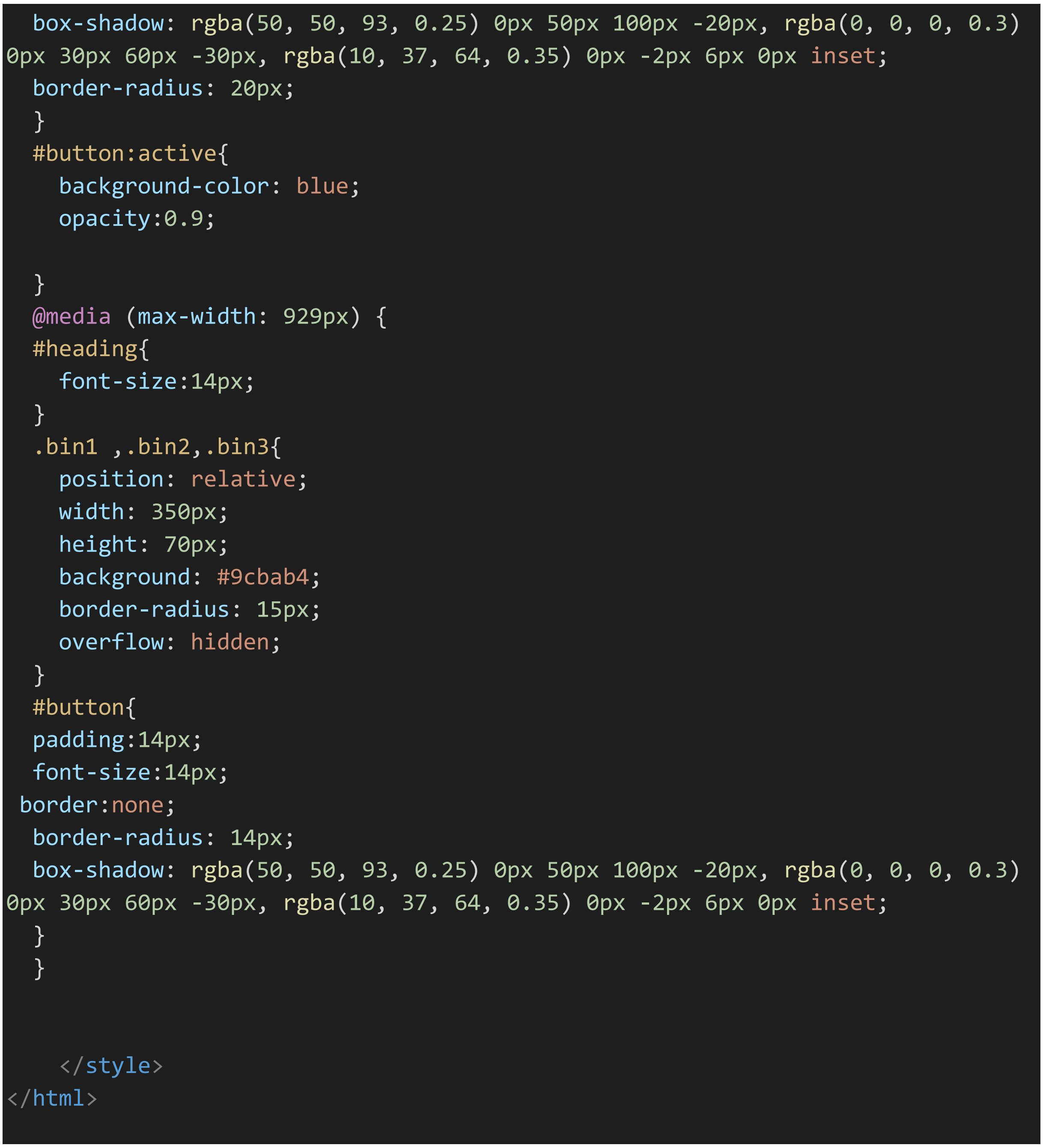
```
  document.querySelector(".text5").innerHTML = `Waste Level : ${value}Cms`;
}
function updateProgressBar2(progressBar, value) {
  value = Math.round(value);
  progressBar.querySelector(".fill2").style.width = `${(value/8)*100}%`;
  progressBar.querySelector(".text2").textContent = `${(value/8)*100}%`;
  document.querySelector(".text4").innerHTML = `Waste Level : ${value}Cms`;

}


let ProgressBar1 = document.querySelector(".bin1");
let ProgressBar2 = document.querySelector(".bin2");


/* Example */
updateProgressBar(ProgressBar1, 5);
updateProgressBar2(ProgressBar2 ,5);

let btn =document.getElementById("button");
btn.addEventListener('click',SelectData);

function setTime()
{
var d = new Date();
var Hour = d.getHours();
var minute = d.getMinutes();
var second = d.getSeconds();
document.getElementById("clock").textContent
=String(Hour)+':'+String(minute)+':'+String(second);}
setInterval(function() {

    setTime();
}, 1000);

</script>

<style>

   .text5, .text4{
     margin: 10px;
    display: block;
    color:black;
    font-size: 20px;
    text-align: center;
  }

  #bin{
    color: black;
```



```
    font-size:30px;
   display: block;
   font-weight:700%;
   padding:20px;
   margin:20px;
   text-align:center;
  }
  #heading{
  color:black;
  margin-bottom: 30px;
  display: block;
  font-size:20px;
  font-weight:700%;
  text-align:center;
  border:solid 2px #75044a;
  padding: 4px;
  border-radius: 20px;
  box-shadow: 2px 3px #75044a;
  }
  html{
    background: linear-gradient(to right top, #051937, #004d7a, #008793,
#00bf72, #a8eb12) no-repeat center center fixed;
    -webkit-background-size: cover;
    -moz-background-size: cover;
    -o-background-size: cover;
    background-size: cover;

  }#bin{
    color: black;
    font-size:30px;
   display: block;
   font-weight:700%;
   padding:20px;
   margin:20px;
   text-align:center;
  }

  html{
    background: #E6C1A6;
  }

  #centerbin1{
    display: flex;
    justify-content: center;
  }
  .bin1 ,.bin2,.bin3{
    position: relative;
    width: 500px;
```



```
    height: 100px;
    background: #9cbab4;
    border-radius: 15px;
    overflow: hidden;
    box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px, rgba(0, 0, 0,
0.3) 0px 30px 60px -30px, rgba(10, 37, 64, 0.35) 0px -2px 6px 0px inset;
  }

  .fill1 ,.fill2,.fill3{
    width: 0%;
    height: 100%;
   background-color: #75044a;
    transition: all 1.2s;
    color: white;

  }
  .box{
    background-color: ghostwhite;
    padding: 50px;
    border-radius: 20px;
    box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px, rgba(0, 0, 0,
0.3) 0px 30px 60px -30px, rgba(10, 37, 64, 0.35) 0px -2px 6px 0px inset;
  }
  .text1,.text2 ,.text3{
    position: absolute;
    top: 50%;
    right: 5px;
    transform: translateY(-50%);
    font: bold 14px "Quicksand", sans-serif;
    color: white;
  }
  #clock{
    font-size:30px;
    color: black;
    display: block;
    text-align:center;
    padding:40px;
  }
  #centerbutton{
    display: flex;
    justify-content: center;
  }
  #button{
  color: white;
  background:#75044a;
  border: none;
  padding:20px;
  font-size:20px;
```

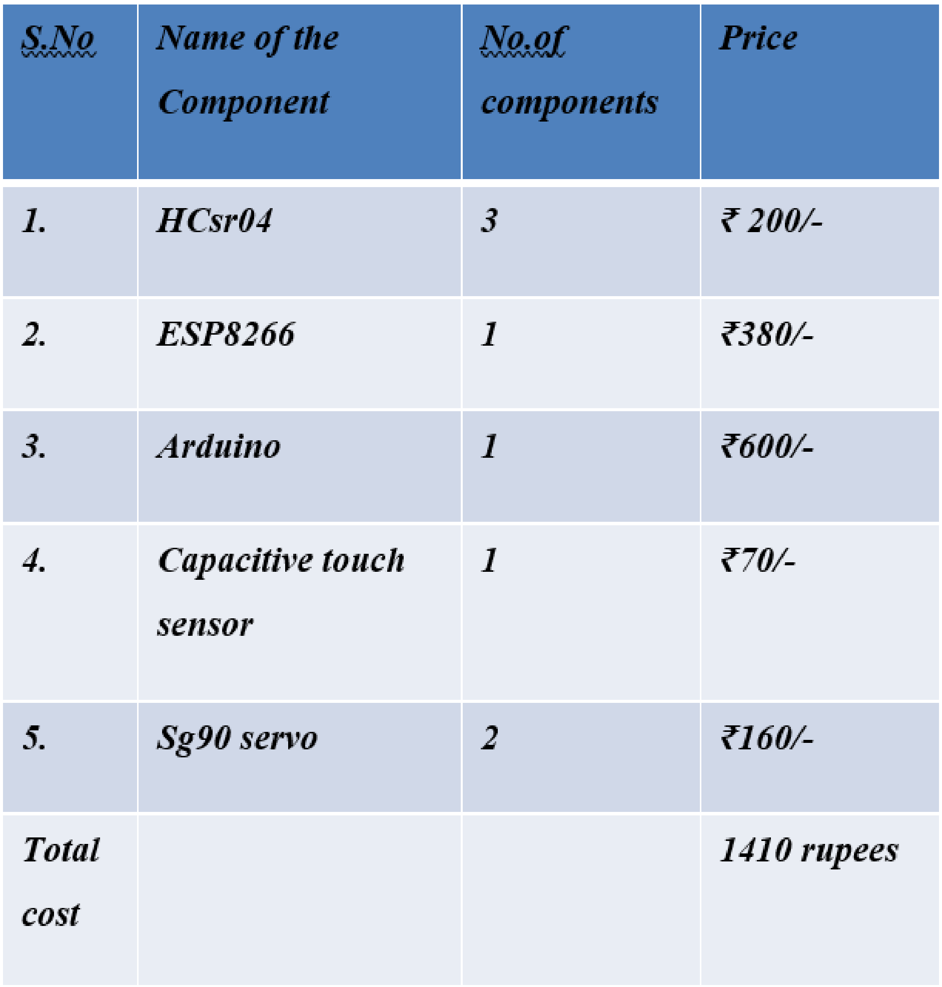


```
  box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px, rgba(0, 0, 0, 0.3)
0px 30px 60px -30px, rgba(10, 37, 64, 0.35) 0px -2px 6px 0px inset;
  border-radius: 20px;
  }
  #button:active{
    background-color: blue;
    opacity:0.9;

  }
  @media (max-width: 929px) {
  #heading{
    font-size:14px;
  }
  .bin1 ,.bin2,.bin3{
    position: relative;
    width: 350px;
    height: 70px;
    background: #9cbab4;
    border-radius: 15px;
    overflow: hidden;
  }
  #button{
  padding:14px;
  font-size:14px;
 border:none;
  border-radius: 14px;
  box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px, rgba(0, 0, 0, 0.3)
0px 30px 60px -30px, rgba(10, 37, 64, 0.35) 0px -2px 6px 0px inset;
  }
  }



    </style>
</html>
```

CHAPTER 5- Conclusion

5.1 Cost Analysis

Table 2 Cost Analysis



S.No	***Name of the Component***	***No.of components***	***Price***
1.	***HCsr04***	***3***	***₹ 200/-***
2.	***ESP8266***	***1***	***₹380/-***
3.	***Arduino***	***1***	***₹600/-***
4.	***Capacitive touch sensor***	***1***	***₹70/-***
5.	***Sg90 servo***	***2***	***₹160/-***
Total cost			***1410 rupees***

5.2 Summmary

We were able to accomplish

- *Automation of Segregation of the waste into toe bins i.e. dry and wet waste.*

  We have achieved this with the use of soil moisture sensor which senses the wetness of the waste and digital capactive senor which senses the presence of waste and two servo motors which aids us in making the *mechanism to put the waste in correct bin according its characteristiscs all these sensors are controlled by Arduino Uno*

- *Indicate the Filled Capacity in each bin using IOT and APP.*
  We have used Esp8266 IoT module and ultrasonic distance sensors to achieve this purpose. The ultrasonic distance sensors give the amount of space filled in each bin which is then uploaded to a cloud database known as firebase with the help of ESP module. The data from the cloud database can be accessed through our Custom made Mobile App.

References

- **https://create.arduino.cc/projecthub/Varun2905/smart-segregator-that-seperates-dry-and-wet-waste-aec602?ref=search&ref_id=waste%20segregator&offset=0**
- **https://create.arduino.cc/projecthub/jaiprak/control-led-from-web-app-using-esp8266-serial-wifi-module-cdf419**
- **https://firebase.google.com/docs/guides**