EMBEDDED SYSTEMS AND IOT
IOT LAB PROJECT

VIDEO LINK -
https://drive.google.com/drive/folders/1SqiCDvdo4BkJWQFIFNJfLNZSLw0P4hn4?usp=sharing

IDEA : Implementation of an Intelligent Waste Management and Segregation System prototype

TEAM MEMBERS:

Ishaan Seshukumar Pothapragada
Arvind Kumar
Hrishit Bhattacharya
Shreyansh Sachan
Shubham Rao
Sruthi Sree N

NEED OF THE PROJECT:

The principle "Cleanliness is next to godliness" holds true in the realm of technology and waste management as well. Maintaining a clean and organized digital environment, as well as efficiently managing and segregating waste, is crucial for the well-being and optimal functioning of individuals, communities, and systems.

With the introduction of initiatives like the "Swachh Bharat Abhiyaan" in 2014, the responsibility of each citizen to ensure cleanliness and proper waste disposal has become paramount. In parallel, it has become essential for individuals to understand the distinction between wet waste and dry waste for effective waste management and segregation practices.

It is to solve these issues in a smart and efficient way that we have implemented ***'Kachra Mitra™'*** (‘*Kachra*’ in Hindi means ‘Waste’ and ‘*Mitra*’ means ‘Friend’)

***NOVELTY*:**

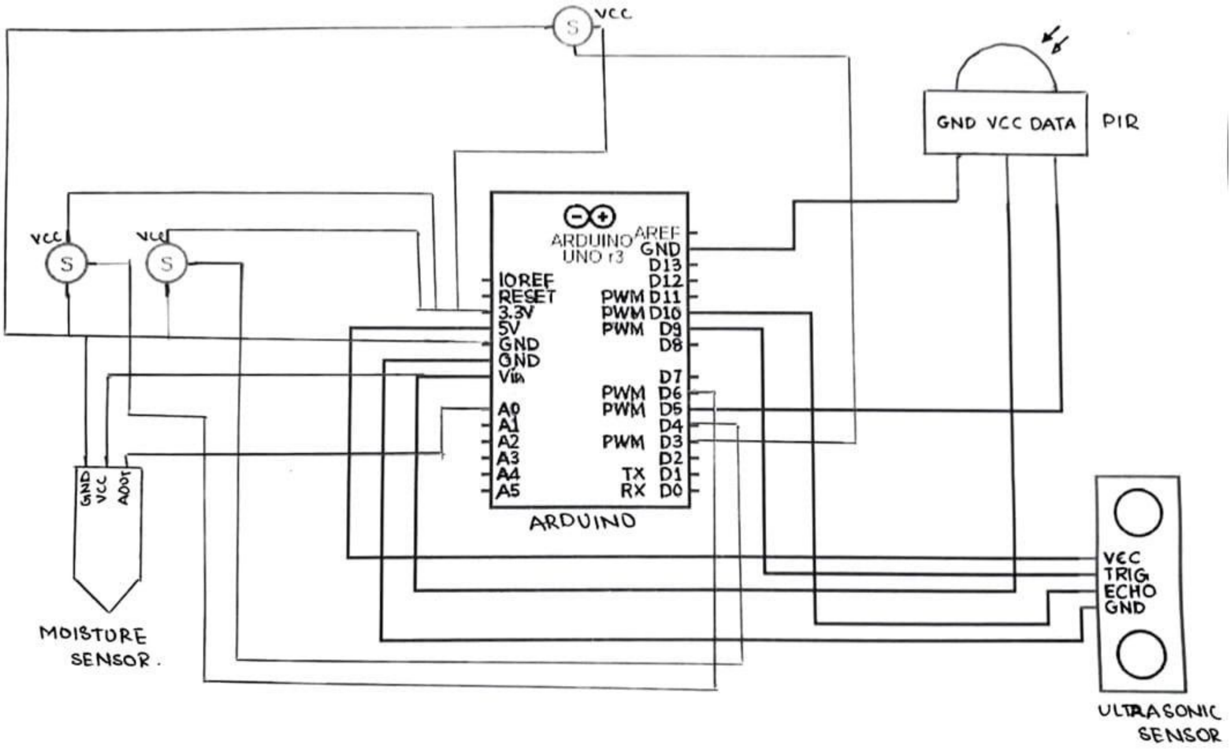
The novelty of our waste management system lies in the **integration of multiple innovative features and technologies** to **automate the segregation and disposal process**. The key aspects that contribute to its novelty are:

1. Automatic waste segregation: Our system employs a tilter mechanism and capacitive soil moisture sensor to achieve intelligent waste segregation. By accurately detecting the moisture level of the waste, the system can effectively separate it into dry and wet sections. This ensures efficient waste management and facilitates appropriate disposal methods for different types of waste.

2. Proximity Sensing Lid System: The implementation of a proximity sensing lid mechanism adds a layer of convenience and hygiene to waste management. By automatically opening the lid when a person is nearby within a particular distance, it eliminates the need for manual lid handling, improving user experience and reducing potential contamination.

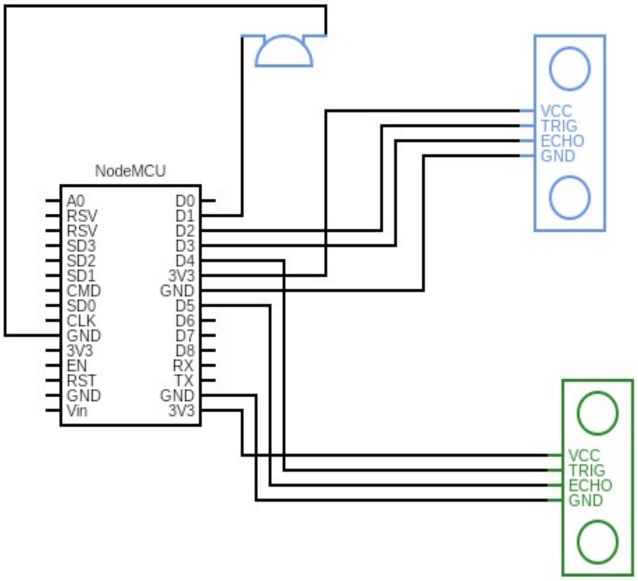
3. Filled-Bin Sensing: The inclusion of ultrasonic sensors to detect if the bin is full brings a proactive approach to waste management. By monitoring the state of the dustbin in real-time, the system can provide timely alerts through a buzzer and a whatsapp message to the user as well, preventing overflow and enabling prompt waste disposal.

CIRCUIT DIAGRAM:



Circuit Diagram for nodeMCU

I



***IMPLEMENTATION*:**

Our waste management system incorporates several innovative features to automate the segregation and disposal process. It utilizes advanced sensors and mechanisms to ensure efficient waste management and improve overall sustainability.

1. Proximity Sensing Lid System

2. Smart Automatic Segregation:

3. Filled-Bin Sensing

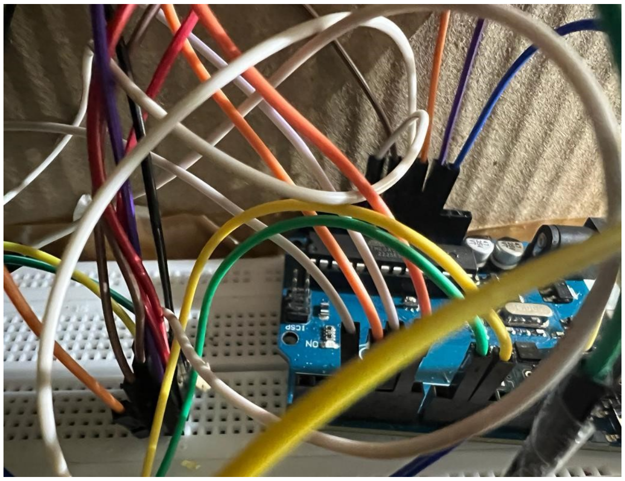
4. Pushing notification to user’s Whatsapp.

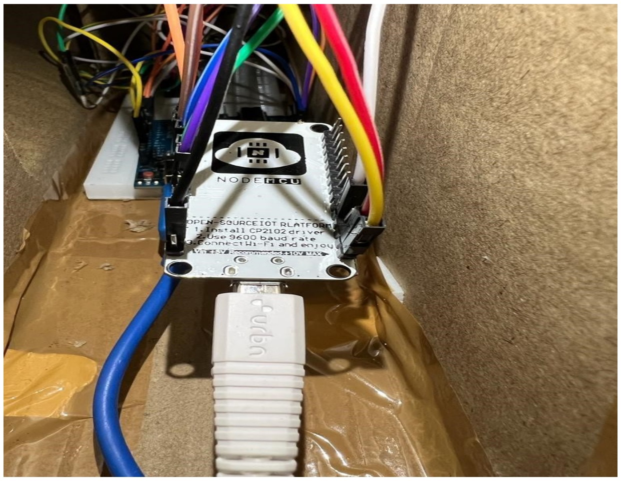
The implementation of these features involves integrating various hardware components and programming logic. **The proximity sensing lid system, consisting of an ultrasonic sensor and a PIR motion sensor and a servo motor** detects the presence of a person and triggers the lid opening mechanism.

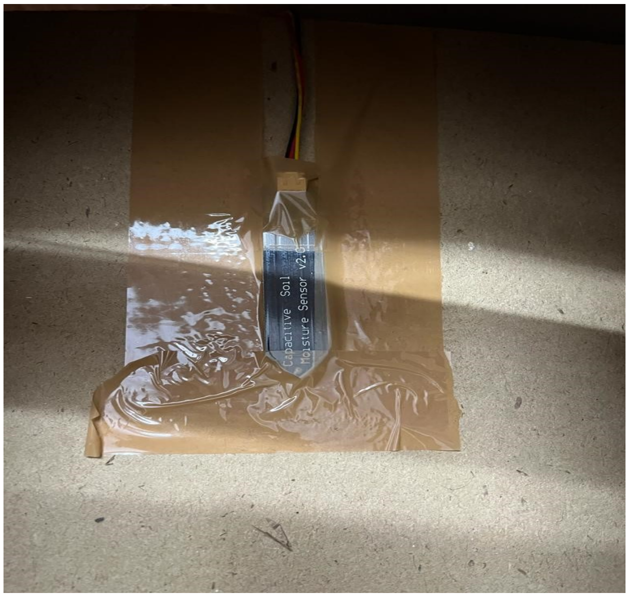
The tilter mechanism on dropping the waste utilizes **the capacitive soil moisture sensor** to detect the moisture level. Based on the moisture levels, the mechanism intelligently directs the waste into the appropriate sections for dry and wet waste.

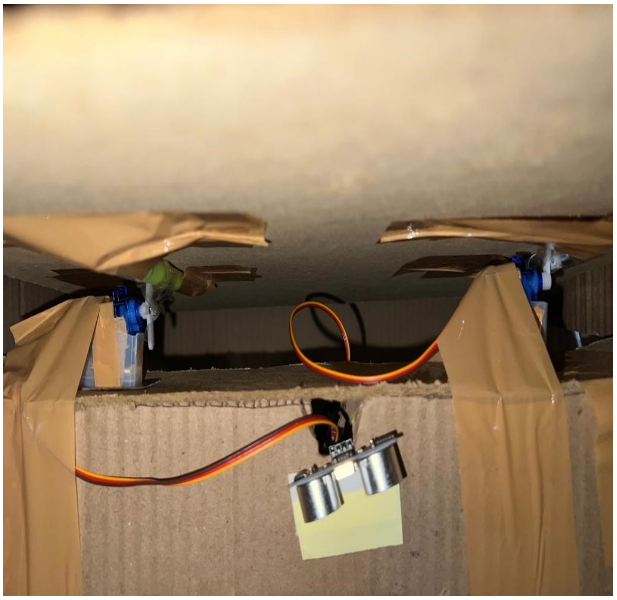
The filled-bin sensing mechanism utilizes the **programmed ultrasonic sensors connected to the NodeMCU.** When the waste reaches the threshold level, the ultrasonic sensors detect that the bin is full and activate the buzzer to signal that the bin requires emptying. This sends a ‘**Whatsapp Message**’ notification to us indicating that the bin requires emptying.

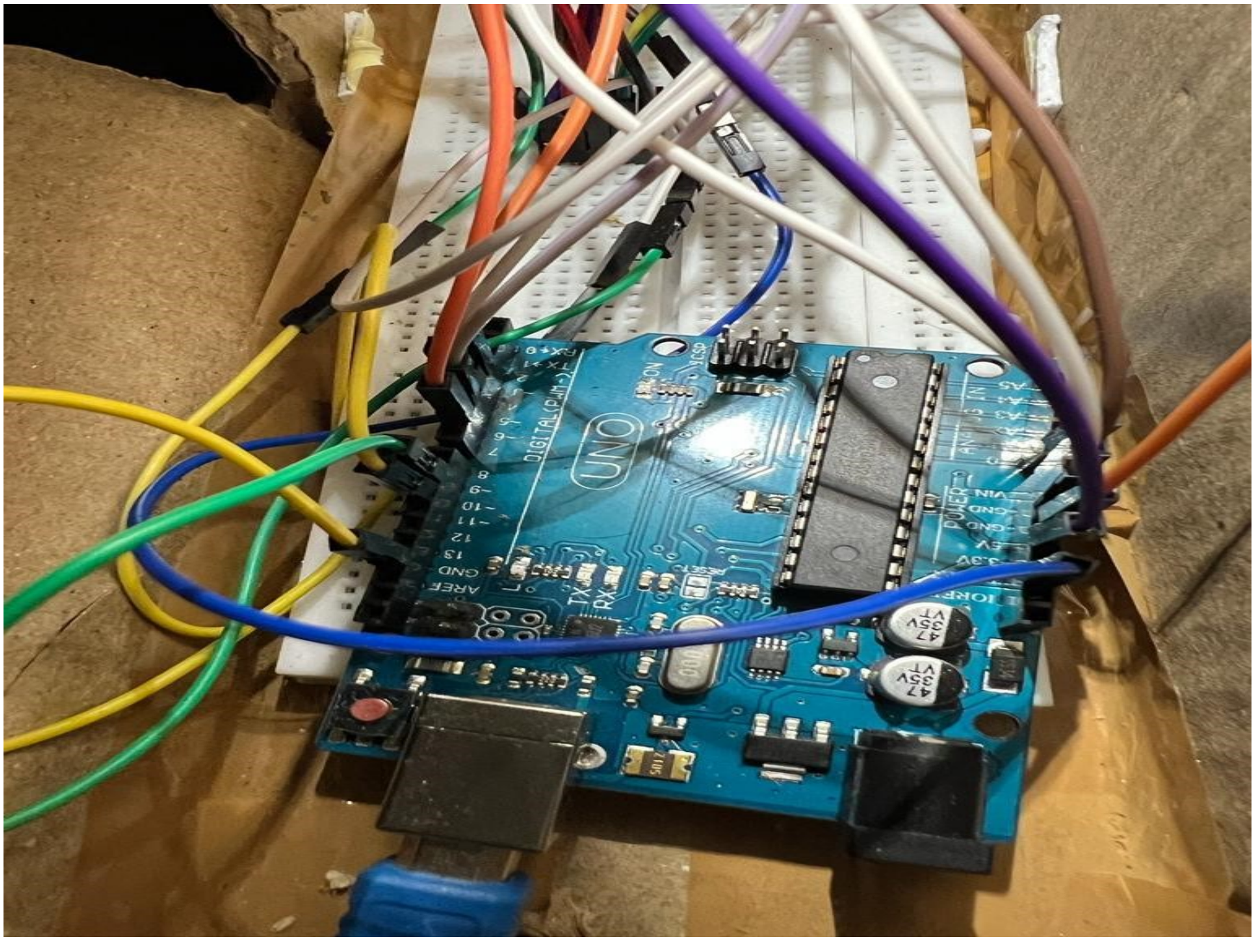
***PHOTOGRAPH OF THE CIRCUIT*:**











KEY-LEARNINGS FROM THE PROJECT:

This project provided us with an ideal platform to **explore the world of IoT** and delve into its intricacies. Over a week-long duration, we meticulously employed a diverse array of instruments, boards, and sensors to bring our waste management system to fruition.

However, the most profound realization we gained from this endeavor was the **critical role of teamwork**. Without effective collaboration and seamless coordination among team members, the successful completion of this project would have been unattainable. Each individual's unique expertise and perspectives contributed to the design, implementation, and troubleshooting phases, emphasizing the importance of collective effort and the synergy it generates.
It provided us with **valuable technical insigh**ts and a profound appreciation for the **vast possibilities** that IoT presents in revolutionizing waste management systems.

CODE:

For Arduino

```
#include <Servo.h>
Servo servo1; // Create object for Servo motor 1
Servo servo2; // Create object for Servo motor 2
Servo servo3; // Create object for Servo motor 3
int position = 0; // Variable to store the position
const int trigPin =9;
const int echoPin =10;
int sensorPin = 5; // PIR Outpin
const int moisturePin = A0; // Analog pin connected to the sensor
int state = 0; // PIR status
long duration;
int distance;
void setup()
{
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(sensorPin, INPUT);
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
  servo1.attach(3); // Set PWM pin 3 for Servo motor 1 lid
  servo2.attach(4); // Set PWM pin 4 for Servo motor 2 tilter
```

```
  servo3.attach(6); // Set PWM pin 4 for Servo motor 2 tilter


}
void loop()
{
  state = digitalRead(sensorPin);
  Serial.print("State: ");
  Serial.print(state);
  Serial.println("");
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2.0;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);


  if (distance < 30 && state == HIGH){


    for(int i=45;i<=150;i+=5)
      {
        servo1.write(i); // Set position of Servo motor 1
        delay(50);
      }
    delay(5000);


    int sensorValue = analogRead(moisturePin);
    Serial.print("Moisture Level :");
    Serial.print(sensorValue);
    Serial.println();


    servo2.write(90);

```

```
if(sensorValue > 550){
  for(int i=90;i>=45;i-=5)
  {
    servo2.write(i); servo3.write(i);

    delay(50);
  }
  delay(2000);
  for(int i=45;i<=90;i+=5)
  {
    servo2.write(i); servo3.write(i);

    delay(50);
  }
}
else{
  for(int i=90;i<=135;i+=5)
  { servo2.write(i);

    servo3.write(i);
  }
  delay(2000);
  for(int i=135;i>=90;i-=5)
  {
    servo2.write(i); servo3.write(i);

  }
}
for(int i=150;i>=45;i-=5)
```

```
    {
      servo1.write(i); // Set position of Servo motor 1
      delay(50);
    } // Set position of Servo motor 1
}
// delay(2000);


  // Serial.print("Moisture: ");
  // delay(1000); Delay between readings



}
```

For NodeMCU

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <UrlEncode.h>

// ----------- Enter your Wi-Fi Details -----------
const char* ssid = "ishu";          // SSID
const char* password = "12345678"; // Password
// ------------------------------------------------


const int buzzer = D1;

const int triggerPin1 = D2;   // Trigger pin for sensor 1
const int echoPin1 = D3;      // Echo pin for sensor 1


const int triggerPin2 = D4;   // Trigger pin for sensor 2
const int echoPin2 = D5;      // Echo pin for sensor 2
const int ChannelField1 = 1;   // Which channel to write data for sensor 1
const int ChannelField2 = 2;   // Which channel to write data for sensor 2


const int notificationThreshold = 3; // Distance threshold for triggering
notification (in cm)


// WhatsApp configuration
String phoneNumber = "+918186045981"; // Your phone number
String apiKey = "4904349";               // CallMeBot API key


void sendMessageToWhatsApp(String message) {
  // Data to send with HTTP POST
  String url = "http://api.callmebot.com/whatsapp.php?phone=" +
phoneNumber + "&apikey=" + apiKey + "&text=" + urlEncode(message);
  WiFiClient client;
  HTTPClient http;
  http.begin(client, url);

  // Specify content-type header
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");


  // Send HTTP POST request
  int httpResponseCode = http.POST(url);
  if (httpResponseCode == 200) {
    Serial.print("Message sent successfully");
  } else {
    Serial.println("Error sending the message");
    Serial.print("HTTP response code: ");
    Serial.println(httpResponseCode);
  }

  // Free resources
  http.end();
}


long pulseInMultiple(int pin, int timeout = 20000) {
  long pulseDuration = pulseIn(pin, HIGH, timeout);
  return pulseDuration;
}

void setup() {
  Serial.begin(9600);

  pinMode(buzzer, OUTPUT);
  pinMode(triggerPin1, OUTPUT);
  pinMode(echoPin1, INPUT);

  pinMode(triggerPin2, OUTPUT);
  pinMode(echoPin2, INPUT);

  WiFi.mode(WIFI_STA);

  // Connect to Wi-Fi
  Serial.print("Connecting to WiFi...");
```

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");
  }

  Serial.println("\nConnected to WiFi");

}



void loop() {

  // Sensor 1

  digitalWrite(triggerPin1, LOW);
  delayMicroseconds(2);

  digitalWrite(triggerPin1, HIGH);

  delayMicroseconds(10);

  digitalWrite(triggerPin1, LOW);

  long duration1 = pulseInMultiple(echoPin1);

  float distance1 = duration1 * 0.034 / 2;


  // Sensor 2

  digitalWrite(triggerPin2, LOW);

  delayMicroseconds(2);

  digitalWrite(triggerPin2, HIGH);

  delayMicroseconds(10);

  digitalWrite(triggerPin2, LOW);

  long duration2 = pulseInMultiple(echoPin2);
  float distance2 = duration2 * 0.034 / 2;



  Serial.print("Distance Sensor 1 (cm): ");

  Serial.println(distance1);

  Serial.print("Distance Sensor 2 (cm): ");
  Serial.println(distance2);


  if (distance1 < notificationThreshold || distance2 <
notificationThreshold) {
    tone(buzzer, 5000);
    noTone(buzzer);
    delay(1500);
    tone(buzzer, 1800);
    delay(1500);
    noTone(buzzer);
```

```
    delay(1000);


    // Send message to WhatsApp
    String message = "I am full please clear me.";
    sendMessageToWhatsApp(message);


    Serial.println("WhatsApp message sent");


    delay(1000);
  }
}
```