Mini project on Garbage Segregator and Bin Level Indicator

Introduction :

Effective waste management is crucial for maintaining a clean and healthy environment. The project presented here aims to improve the process of waste segregation and bin level indication by employing smart technologies such as capacitive and inductive sensors, servo motors, an ESP-32 microcontroller, and a load sensor. This smart bin can differentiate between types of waste and segregate them accordingly while monitoring the bin's fill level**.**

Objectives :

The primary objective of this project is to develop an intelligent garbage segregation and bin level indication system that improves waste management efficiency. By integrating capacitive and inductive proximity sensor , servo motor, and an ESP-32 microcontroller, the system can automatically classify and sort waste based on the type of material detected. Additionally, the system uses a load sensor to monitor the bin fill level and sends data to the ThingSpeak web application for real time monitoring.This automation aims to optimize the waste sorting process, facilitate proper recycling, and provide valuable data for wate management.

Components Required :

- **ESP-32 Microcontroller -** For controlling and processing input from sensors and actuators.
- **Capacitive Proximity Sensor -** Detects paper, glass, and plastic waste.
- **Inductive Proximity Sensor -** Detects metal waste.
- **Servo Motors -** Two motors control the hatch doors for waste segregation.
- **Load Sensor -** Measures the weight of waste in the bin.
- **ThingSpeak Web Application** - Online platform for displaying real-time data.

Working Principle :

1. Waste Input: When waste is placed in the smart bin, the system identifies the type of material using proximity sensors.

2. Material Identification:

- The capacitive proximity sensor detects paper, glass, and plastic waste.
- The inductive proximity sensor detects metal waste.

3. Hatch Door Control :

- If the capacitive proximity sensor detects suitable waste, servo motor 1 operates to open and close a hatch door for the designated waste compartment.

- If the inductive proximity sensor detects metal waste, servo motor 2 operates to open and close a different hatch door for the metal compartment.

4. Load Monitoring : The load sensor measures the weight of the waste in the bin and sends data to the ESP-32 microcontroller.

5. Data Display: The ESP-32 microcontroller transmits data to the ThingSpeak web application for real-time monitoring of bin status.

Code Implementation :

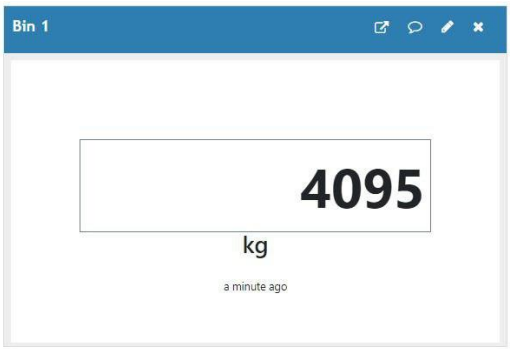
```
#include <ESP32Servo.h>

#include <Wire.h>

#include "HX711.h"

#include <WiFi.h>

#include <ThingSpeak.h>


Servo inductiveServo;

Servo capacitiveServo;

HX711 scale;
```

```
const int inductiveSensorPin = 34;
const int capacitiveSensorPin = 35;
const int inductiveThreshold = 500;
const int capacitiveThreshold = 500;

const int LOADCELL_DOUT_PIN = 26;
const int LOADCELL_SCK_PIN = 27;

const char *ssid = "Vivo v21e";
const char *password = "blackbeast";
const char *thingSpeakApiKey = "6YMLB2YG4X50ZUNY";
const int thingSpeakChannelID =2497312;

WiFiClient client;

void setup() {
  Serial.begin(9600);
  inductiveServo.attach(32); // Attach servo to GPIO pin 32
  capacitiveServo.attach(33); // Attach servo to GPIO pin 33
  pinMode(inductiveSensorPin, INPUT);
  pinMode(capacitiveSensorPin, INPUT);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  scale.set_scale();
  scale.tare();

  WiFi.begin(ssid, password);
  Serial.print("Connecting to ");
  Serial.println(ssid);
```

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

  ThingSpeak.begin(client); // Initialize ThingSpeak
}
void loop() {
  int inductiveSensorValue = analogRead(inductiveSensorPin);
  int capacitiveSensorValue = analogRead(capacitiveSensorPin);
  float weight = scale.get_units(10);

  if (inductiveSensorValue > inductiveThreshold) {
    inductiveServo.write(90); // Rotate servo 90 degrees clockwise
    delay(1000); // Wait for 1 second
    inductiveServo.write(0);
    delay(1000); // Wait for 1 second
  }
 // Check if capacitive sensor detects plastic or other materials
  if (capacitiveSensorValue > capacitiveThreshold) {
    capacitiveServo.write(90);
    delay(1000);
    capacitiveServo.write(0);
    delay(1000);
  }
  ThingSpeak.writeField(thingSpeakChannelID, 1, inductiveSensorValue, thingSpeakApiKey);
  ThingSpeak.writeField(thingSpeakChannelID, 2, capacitiveSensorValue, thingSpeakApiKey);
```

```
  ThingSpeak.writeField(thingSpeakChannelID, 3, weight, thingSpeakApiKey);
  ThingSpeak.writeField(thingSpeakChannelID, 4, weight, thingSpeakApiKey); // Sending weight to two fields for redundancy

  delay(20000); // Send data to ThingSpeak every 20 seconds
}
```

To upload this code to the ESP-32 microcontroller, you can use the Arduino Integrated Development Environment (IDE). The Arduino IDE is a popular and user-friendly software tool for programming and uploading code to a variety of microcontrollers, including the ESP-32.



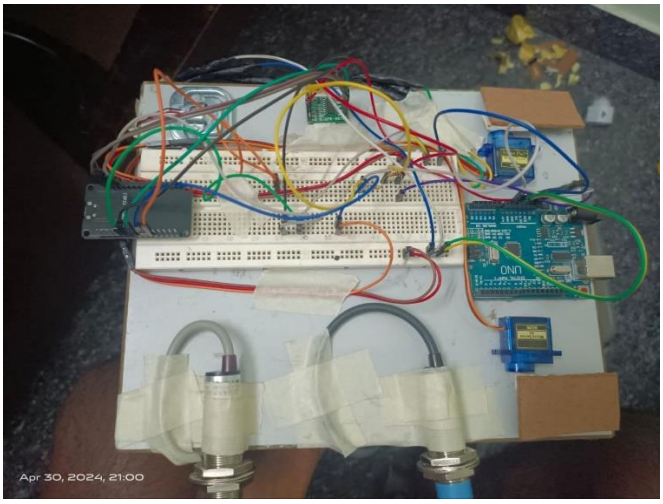
From the above figure shows that weight of garbages in a bin through ThingSpeak Web application.

Project Implementation :

Step 1 : Setup: Mount the sensors, servo motors, and load sensor on the bin.

Step 2 : Wiring and Connections: Connect the sensors and actuators to the ESP-32 microcontroller according to the circuit design.

Step 3 : Programming: Write the code for the ESP-32 microcontroller to control the sensors, servo motors, and load sensor.



Step 4 : Data Integration: Configure the ESP-32 to send data to the ThingSpeak web application for monitoring and visualization.

Advantages :

- **Improved Efficiency:** Automated sorting reduces manual intervention and speeds up the waste management process.
- **Real-Time Monitoring:** Continuous tracking of bin status allows for timely emptying and maintenance.
- **Cost-Effective:** By separating waste at the source, it becomes easier to manage and recycle materials, potentially saving money.
- **Environmentally Friendly:** Proper segregation enhances recycling efforts and reduces landfill waste.
- **Data-Driven Insights:** Real-time data provides valuable insights into waste generation patterns**.**