#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "YOUR\_WIFI\_SSID";

const char\* password = "YOUR\_WIFI\_PASSWORD";

const char\* mqtt\_server = "MQTT\_BROKER\_IP\_ADDRESS";

const char\* client\_id = "garbage\_bin";

const char\* garbage\_topic = "garbage\_level";

WiFiClient espClient;

PubSubClient client(espClient);

const int trigPin = D1;

const int echoPin = D2;

void setup\_wifi() {

delay(10);

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");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void callback(char\* topic, byte\* payload, unsigned int length) {

// Handle incoming messages from MQTT broker if needed

}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

if (client.connect(client\_id)) {

Serial.println("connected");

client.subscribe(garbage\_topic);

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

void setup() {

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

long duration, distance;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

Serial.print("Distance: ");

Serial.println(distance);

// Use machine learning model to predict garbage level

// Publish garbage level to MQTT broker

char garbage\_message[10];

snprintf(garbage\_message, 10, "%ld", distance);

client.publish(garbage\_topic, garbage\_message);

delay(2000); // Adjust delay as needed

}