SPRINT - 2

Date	5 NOV 2022
Team ID	PNT2022TMID01880
Project Name	Project- About Smart Waste Management System For Metropolitan Cities

CODE FOR DETECTING BIN LEVEL AND DISPLAYING IT IN IBM CLOUD:

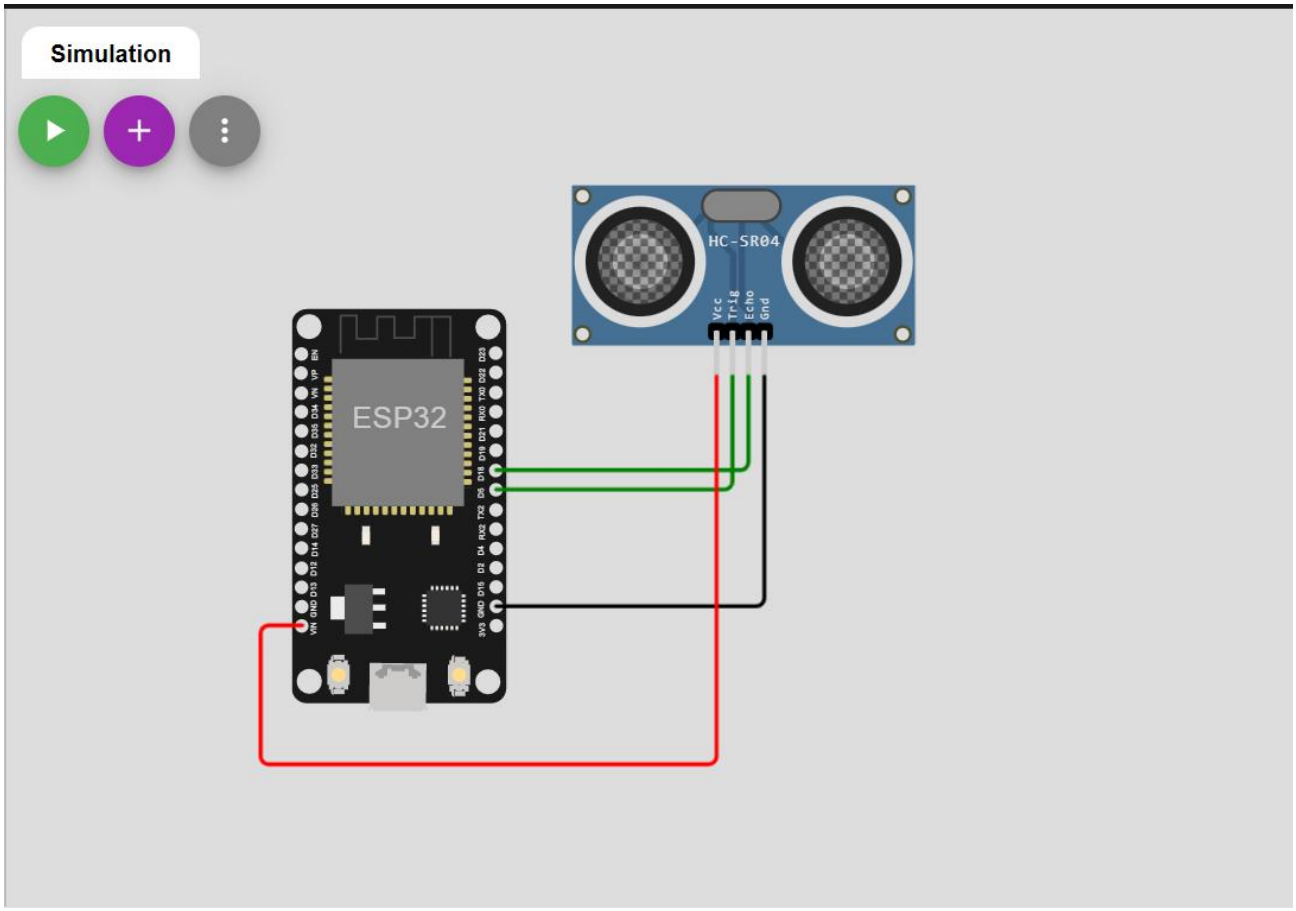
esp32-ultraSonic.ino:

```
// Project: Smart Waste Management System for Metropolitan cities
#Team ID: PNT2022TMID01880
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";
#define ORG "1hx03x"
#define DEVICE_TYPE "cloud"
#define DEVICE_ID "232323"
#define TOKEN "12345678"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

const int trigPin = 5;
const int echoPin = 18;
long duration;
float distance;
```

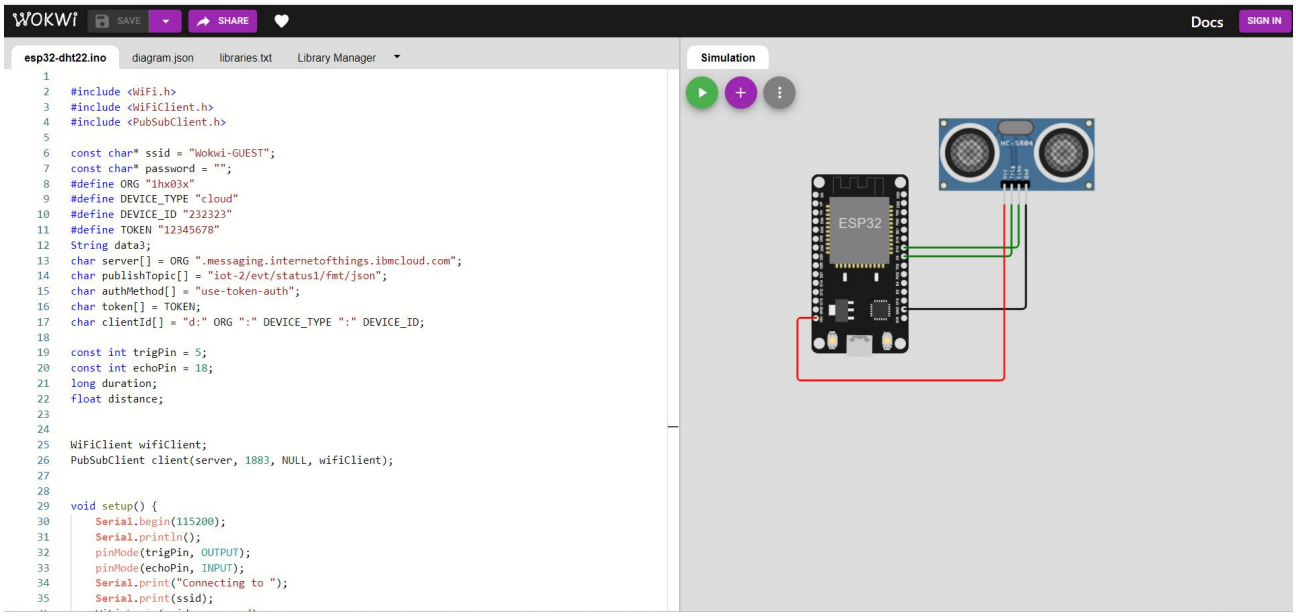
```
WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);
void setup() {
    Serial.begin(115200);
    Serial.println();
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.print("Connecting to ");

    Serial.print(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");

    Serial.print("WiFi connected, IP address: ");
    Serial.println(WiFi.localIP());

    if (!client.connected()) {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        Serial.println("Bluemix connected");
    }
}
void loop() {
    client.loop();
    digitalWrite(trigPin, LOW); delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); delayMicroseconds(10);
    digitalWrite(trigPin, LOW); duration = pulseIn(echoPin, HIGH);
```

```
    distance = duration * 0.034/2;
    PublishData(distance);
    delay(5000);
    }


void PublishData(float dist) {
int binlevel = map(dist, 0 , 400, 100,0);
Serial.println(binlevel);
String payload = "{\"Distance\":";
payload += dist;
payload += ",\"latitude\":""\"11.5215151\"";
payload += ",\"longitude\":""\"77.1388062\"";
payload += ",\"binlevel\":";


payload += binlevel;
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*)
payload.c_str())) {
Serial.println("Publish ok");
}

 else {
Serial.println("Publish failed");
      }
}
```

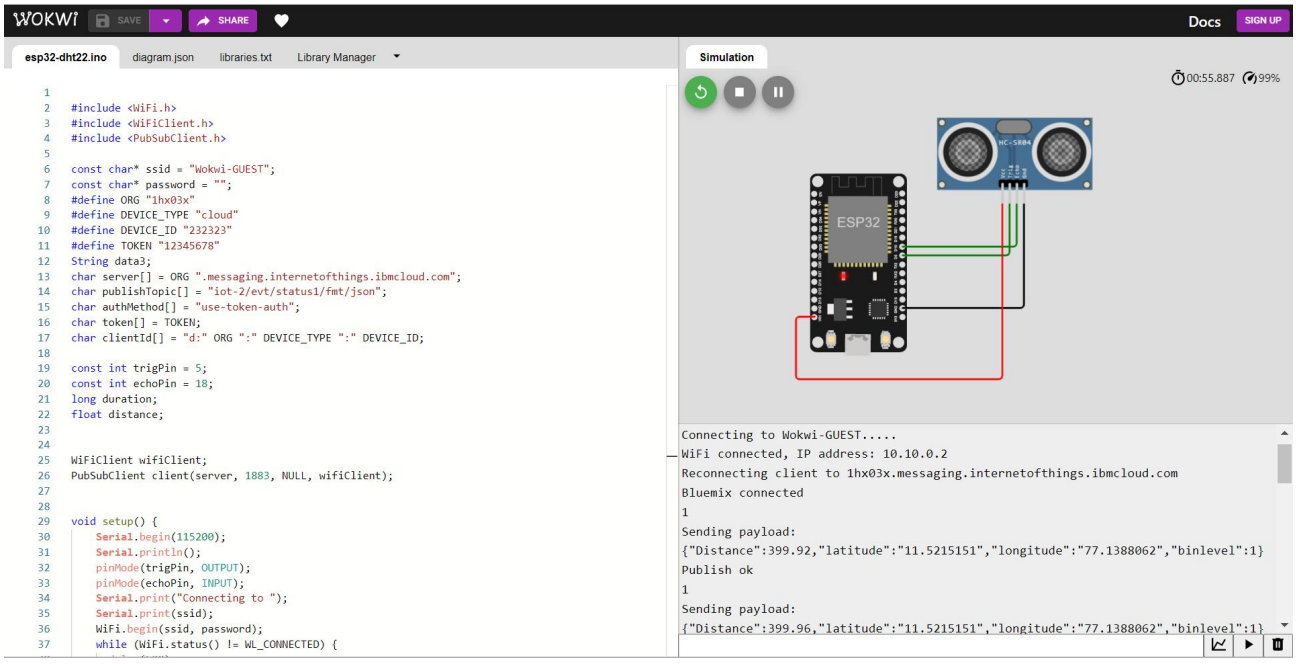
CIRCUIT DIAGRAM:


Simulation
HC-SR04
Vcc
Trig
Echo
Gnd
ESP32

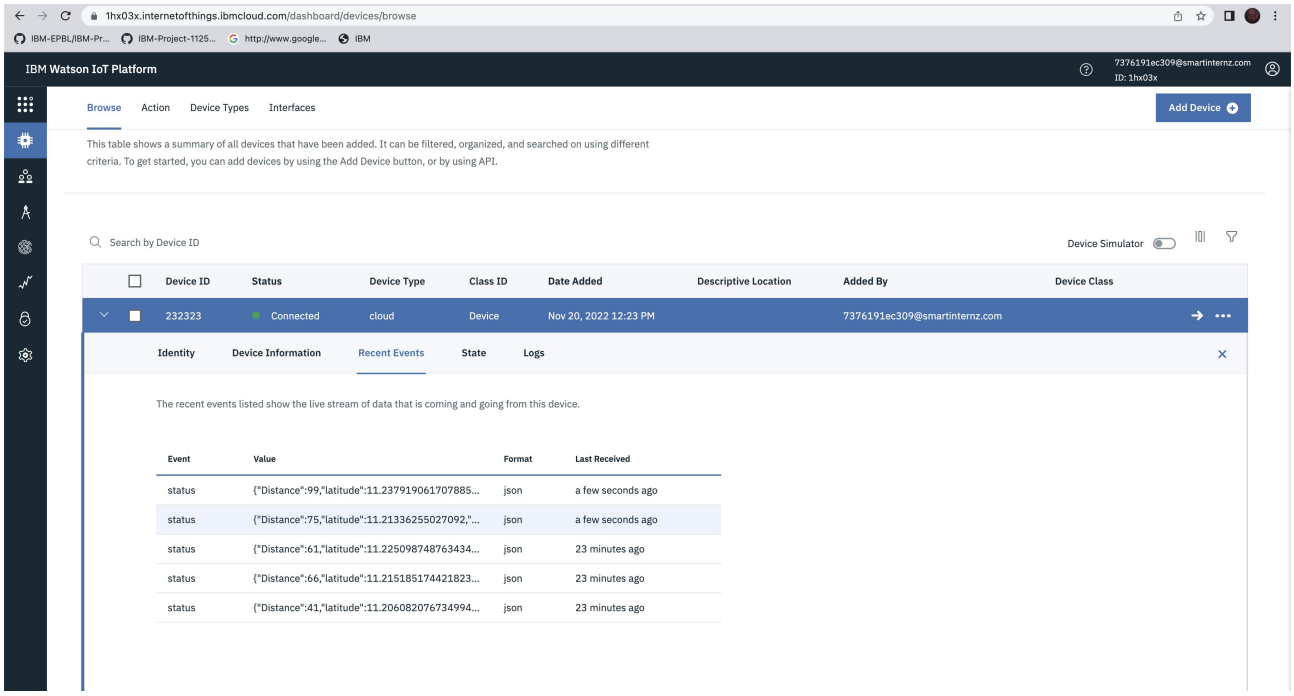
SIMULATION IN WOKWI:




```
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";
#define ORG "1hx03x"
#define DEVICE_TYPE "cloud"
#define DEVICE_ID "232323"
#define TOKEN "12345678"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

const int trigPin = 5;
const int echoPin = 18;
long duration;
float distance;


WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);


void setup() {
    Serial.begin(115200);
    Serial.println();
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.print("Connecting to ");
    Serial.print(ssid);
```




```
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";
#define ORG "1hx03x"
#define DEVICE_TYPE "cloud"
#define DEVICE_ID "232323"
#define TOKEN "12345678"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

const int trigPin = 5;
const int echoPin = 18;
long duration;
float distance;


WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);


void setup() {
    Serial.begin(115200);
    Serial.println();
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.print("Connecting to ");
    Serial.print(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
```

WOKWI LINK: https://wokwi.com/projects/348836138877715027

IBM WATSON IOT PLATFORM OUTPUT: