

# FINAL CODE AND OUTPUT

Date	15 November 2022
Team ID	PNT2022TMID04392
Project Name	Smart Waste Management System for Metropolitan Cities

## CODING:

```
// #define ORG "3defa"
// #define DEVICE_TYPE "hariprasath"
// #define DEVICE_ID "12345"
// #define TOKEN "CpL-H1C-Pt4i9iM-F5"

#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);
// credentials of IBM Accounts -
#define ORG "3defa" //IBM organisation id
#define DEVICE_TYPE "hariprasath" // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "12345" // Device ID mentioned in ibm watson iot platform
#define TOKEN "CpL-H1C-Pt4i9iM-F5" // Token
// customise above values -

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test
format of strings
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id
```

```

//
WiFiClient wifiClient; // creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);
#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;

void setup()
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir
  pinMode(4, INPUT);
  //ledpins
  pinMode(23, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(15, OUTPUT);

  lcd.init();

  lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print("");
  wifiConnect();
  mqttConnect();
}
float readcmCM()
{
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}
void loop()
{
  lcd.clear();
  publishData();
  delay(500);
  if (!client.loop())

```

```

{
mqttConnect(); //function call to connect to IBM
}
}
/* -retrieving to cloud */ void wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect()
{
if (!client.connected())
{
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while(!client.connect(clientId, authMethod, token))
{
Serial.print("."); delay(500);
}
initManagedDevice(); Serial.println();
}
}
void initManagedDevice()
{
if (client.subscribe(topic))
{
Serial.println("IBM subscribe to cmd OK");
}
else
{
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
float cm = readcmCM();

```

```
if(digitalRead(34)) //PIR motion detection
{
  Serial.println("Motion Detected"); Serial.println("Lid Opened"); digitalWrite(15, HIGH);
}
else
{
  digitalWrite(15, LOW);
}
if(digitalRead(34)== true)
{
  if(cm <= 100) //Bin level detection
  {
    digitalWrite(2, HIGH);
    Serial.println("High Alert!!!,Trash bin is about to be full");
    Serial.println("Lid Closed");
    lcd.print("Full! Don't use");
    delay(2000);
    lcd.clear();
    digitalWrite(4, LOW);
    digitalWrite(23, LOW);
  }
  else if(cm > 150 && cm < 250)
  {
    digitalWrite(4, HIGH);
    Serial.println("Warning!!,Trash is about to cross 50% of bin level");
    digitalWrite(2,LOW);
    digitalWrite(23, LOW);
  }
  else if(cm > 250 && cm <=400)
  {
    digitalWrite(23, HIGH);
    Serial.println("Bin is available");
    digitalWrite(2,LOW);
    digitalWrite(4, LOW);
  }
  delay(10000);
  Serial.println("Lid Closed");
}
else
{
  Serial.println("No motion detected");
}
if(cm <= 100)
{

```

```

digitalWrite(21,HIGH);
String payload = "{\"High Alert!!\":\":";
payload += cm;
payload += "left\" }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to
cloud successfully,prints publish ok or prints publish failed
{
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning!!\":\":";
payload += dist;
payload += "left\" }";
Serial.print("\n");
Serial.print("Sending distance: ");
Serial.println(cm);
if(client.publish(publishTopic,(char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
float inches = (cm / 2.54); //print on LCD lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm, 1);
lcd.setCursor(14,1);
delay(1000);
lcd.clear();
}

```

# SCREENSHOTS:

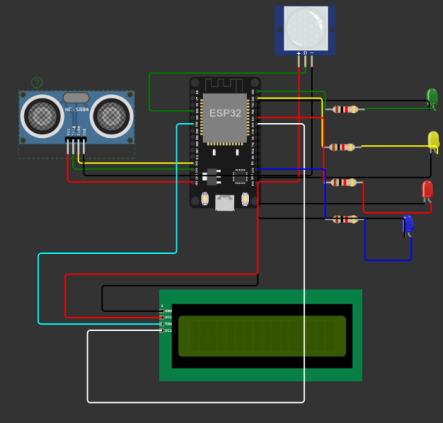
WOKWI

sketch.ino copy

sketch.ino diagram.json libraries.txt Library Manager

```
1
2
3
4
5 // #define ORG "3defsta"
6 // #define DEVICE_TYPE "hariprasath"
7 // #define DEVICE_ID "12345"
8 // #define TOKEN "CpL-H1C-Pt4i9iM-F5"
9
10
11
12 #include <WiFi.h> //library for wifi
13 #include <PubSubClient.h> //library for MQTT
14 #include <LiquidCrystal_I2C.h>
15
16 LiquidCrystal_I2C lcd(0x27, 20, 4);
17 // credentials of IBM Accounts -
18 #define ORG "3defsta" //IBM organisation id
19 #define DEVICE_TYPE "hariprasath" // Device type mentioned in ibm watson iot
20 #define DEVICE_ID "12345" // Device ID mentioned in ibm watson iot platform
21 #define TOKEN "CpL-H1C-Pt4i9iM-F5" // Token
22 // customise above values -
23
24 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server nam
25 char publishTopic[] = "iot-2/evt/data/fmt/json";
26 char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command
27 char authMethod[] = "use-token-auth"; // authentication method
28 char token[] = TOKEN;
```

Simulation



25°C  
Partly cloudy

WOKWI

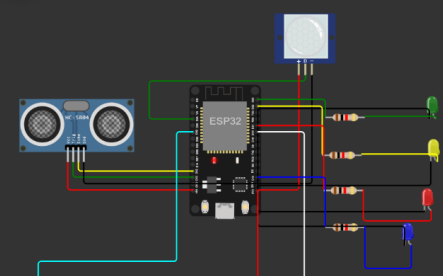
sketch.ino copy

sketch.ino diagram.json libraries.txt Library Manager

```
1
2
3
4
5 // #define ORG "3defsta"
6 // #define DEVICE_TYPE "hariprasath"
7 // #define DEVICE_ID "12345"
8 // #define TOKEN "CpL-H1C-Pt4i9iM-F5"
9
10
11
12 #include <WiFi.h> //library for wifi
13 #include <PubSubClient.h> //library for MQTT
14 #include <LiquidCrystal_I2C.h>
15
16 LiquidCrystal_I2C lcd(0x27, 20, 4);
17 // credentials of IBM Accounts -
18 #define ORG "3defsta" //IBM organisation id
19 #define DEVICE_TYPE "hariprasath" // Device type mentioned in ibm watson iot
20 #define DEVICE_ID "12345" // Device ID mentioned in ibm watson iot platform
21 #define TOKEN "CpL-H1C-Pt4i9iM-F5" // Token
22 // customise above values -
23
24 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server nam
25 char publishTopic[] = "iot-2/evt/data/fmt/json";
26 char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command
27 char authMethod[] = "use-token-auth"; // authentication method
28 char token[] = TOKEN;
```

Simulation

00:01.349 26%



Connecting to Wifi

25°C  
Partly cloudy

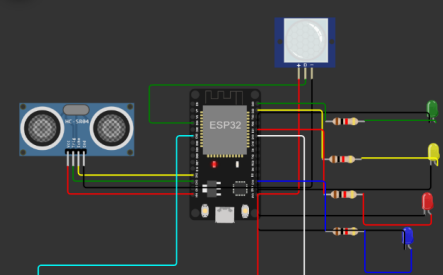
WOKWI

sketch.ino diagram.json libraries.txt Library Manager

```
71 lcd.clear();
72 publishData();
73 delay(500);
74 if (!client.loop())
75 {
76   mqttConnect(); //function call to connect to IBM
77 }
78 }
79 /* -retrieving to cloud */ void wifiConnect()
80 {
81   Serial.print("Connecting to ");
82   Serial.print("Wifi");
83   WiFi.begin("Wokwi-GUEST", "", 6);
84   while (WiFi.status() != WL_CONNECTED)
85   {
86     delay(500);
87     Serial.print(".");
88   }
89   Serial.print("Wifi connected, IP address: ");
90   Serial.println(WiFi.localIP());
91 }
92 void mqttConnect()
93 {
94   if (!client.connected())
95   {
96     Serial.print("Reconnecting MQTT client to ");
97     Serial.println(server);
98     while(!client.connect(clientId, authMethod, token))
```

Simulation

00:26.657 68%



No motion detected  
No motion detected  
No motion detected  
No motion detected  
No motion detected  
No motion detected

25°C  
Partly cloudy

WOKWI

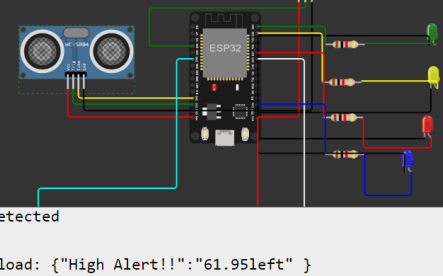
sketch.ino diagram.json libraries.txt Library Manager

```
71 lcd.clear();
72 publishData();
73 delay(500);
74 if (!client.loop())
75 {
76   mqttConnect(); //function call to connect to IBM
77 }
78 }
79 /* -retrieving to cloud */ void wifiConnect()
80 {
81   Serial.print("Connecting to ");
82   Serial.print("Wifi");
83   WiFi.begin("Wokwi-GUEST", "", 6);
84   while (WiFi.status() != WL_CONNECTED)
85   {
86     delay(500);
87     Serial.print(".");
88   }
89   Serial.print("Wifi connected, IP address: ");
90   Serial.println(WiFi.localIP());
91 }
92 void mqttConnect()
93 {
94   if (!client.connected())
95   {
96     Serial.print("Reconnecting MQTT client to ");
97     Serial.println(server);
98     while(!client.connect(clientId, authMethod, token))
```

Simulation

00:50.953 102%

Editing Ultrasonic Distance Sensor  
Distance: 62cm

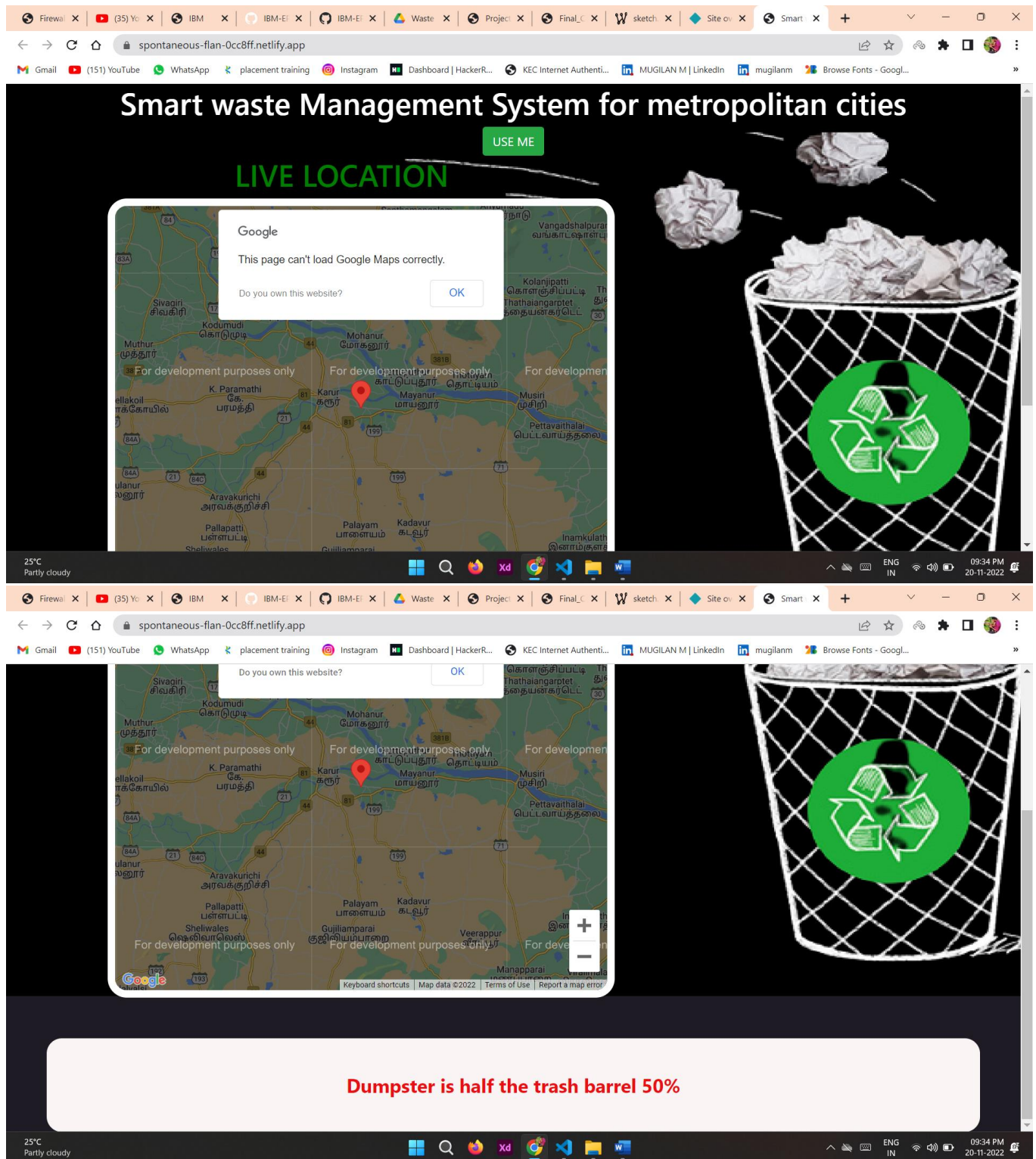


No motion detected

Sending payload: {"High Alert!!":"61.951left" }  
Publish OK

Sending distance: 61.95  
Publish OK

25°C  
Partly cloudy



LINK:

<https://wokwi.com/projects/348861592195039828>