

NAME : HARIRAJ S

REG NO : 210619205014

PROGRAM

Smart Waste Management System for Metropolitan Cities

ASSIGNMENT 4:

Write code and connections in wokwi for ultrasonic sensors. Whenever distance is less than 100 cms

send "alert" to ibm cloud and display in device recent events. Upload document with wokwi share link and images of ibm cloud.

CODE:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "ztcz45"

#define DEVICE_TYPE "naveen"

#define DEVICE_ID "naveen123"

#define TOKEN "123456789"

#define speed 0.034 #define led 14 char server[] = ORG

".messaging.internetofthings.ibmcloud.com"; char

publishTopic[] = "iot-2/evt/Data/fmt/json"; char topic[] = "iot-

2/cmd/home/fmt/String"; char authMethod[] = "use-token-

auth"; char token[] = TOKEN; char clientId[] = "d:" ORG ":"

DEVICE_TYPE ":" DEVICE_ID;

PubSubClient client(server, 1883, wifiClient);
```

```

void publishData();

const int trigpin=5;

const          int
echopin=18; String
command;      String
data="";      long
duration; float dist;

void setup()

{

  Serial.begin(115200);

  pinMode(led, OUTPUT);

  pinMode(trigpin, OUTPUT);

  ...

```

[10:32 pm, 23/10/2022] Gogul B.E CSE:

```

} 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(client.subscribe(
e(topic));

Serial.println("IBM subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}

void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<100){
String payload = "{\Normal
Distance\":"; payload += dist; payload
+= "}";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload); if (client.publish(publishTopic,
(char*) payload.c_str())) {
Serial.println("Publish OK");
}
}

```

```

}

if(dist>101 && dist<111){

String payload = "{\Alert distance\":";

payload += dist; payload += "}";

Serial.print("\n");

Serial.print("Sending      payload:      ");

Serial.println(payload); if(client.publish(publishTopic,

(char*) payload.c_str())) {

Serial.println("Warning  crosses  110cm  --  it  automaticaly  of  the  loop");

digitalWrite(led,HIGH);

}else {

Serial.println("Publish FAILED");

}

}

}

}

void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){

Serial.print("callback    invoked    for

topic:");  Serial.println(subscribeTopic);

for(int i=0; i<payloadLength; i++){ dist +=

(char)payload[i];

}

Serial.println("data:"+ data3);

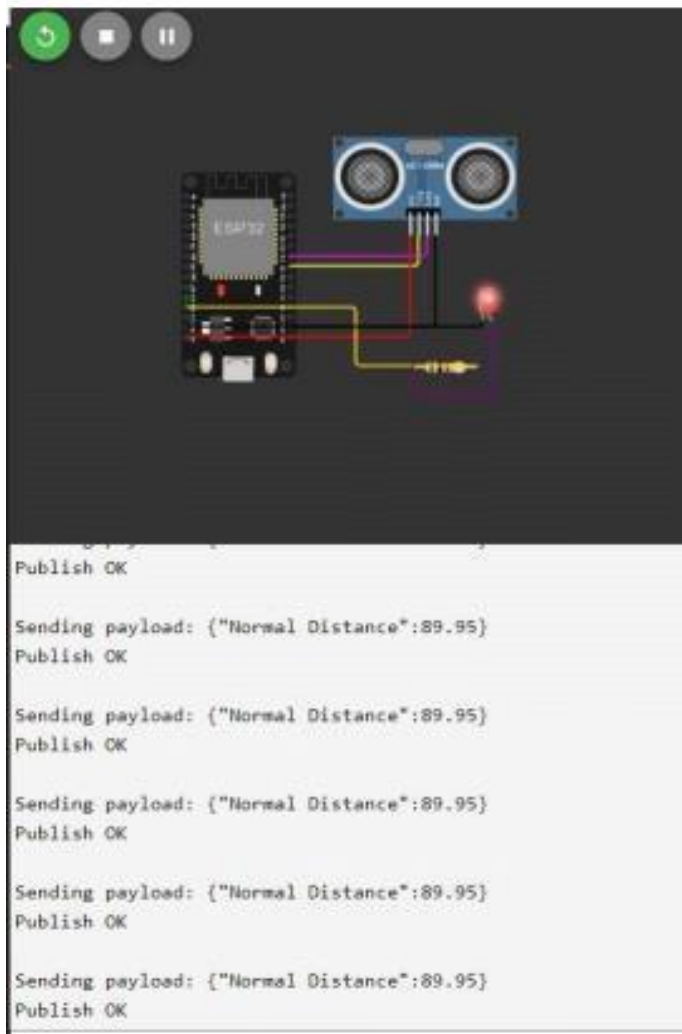
if(data3=="lighton"){

```

output:



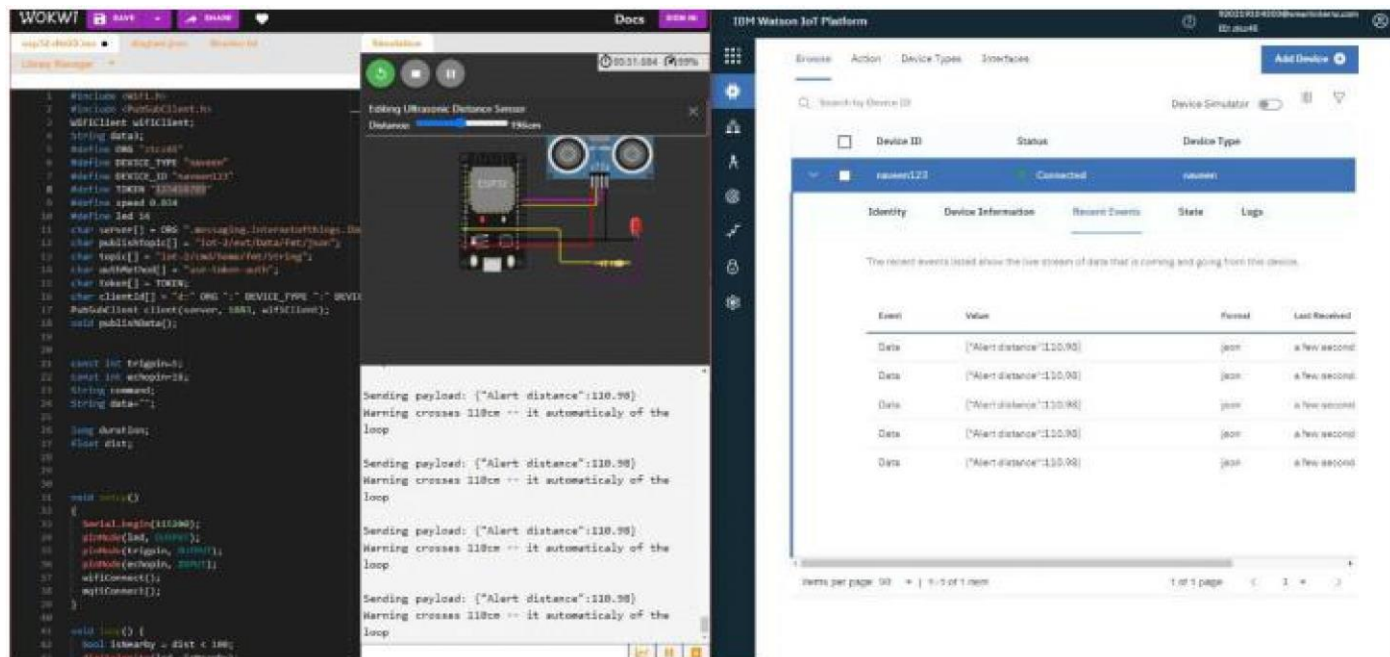
1. When distance under 100 cm it will show normal distance.



2. When distance cross 100 cm it will show ALERT warning message distance

3. When it cross above 110 cm it today move to iff state once it
reduce to 110 it on again

Connection information:



Basic connection information about this device.

Organization ID : ztc45

Device Type : THOL

Device ID : THOL123

Authentication Method : use-token-auth Authentication Token : 123456789

Identity	Device Information	Recent Events	State	Logs
The recent events listed show the live stream of data that is coming and going from this device.				
Event	Value	Format	Last Received	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	