Name:KR.Vashundhara

Reg N0:910619104098

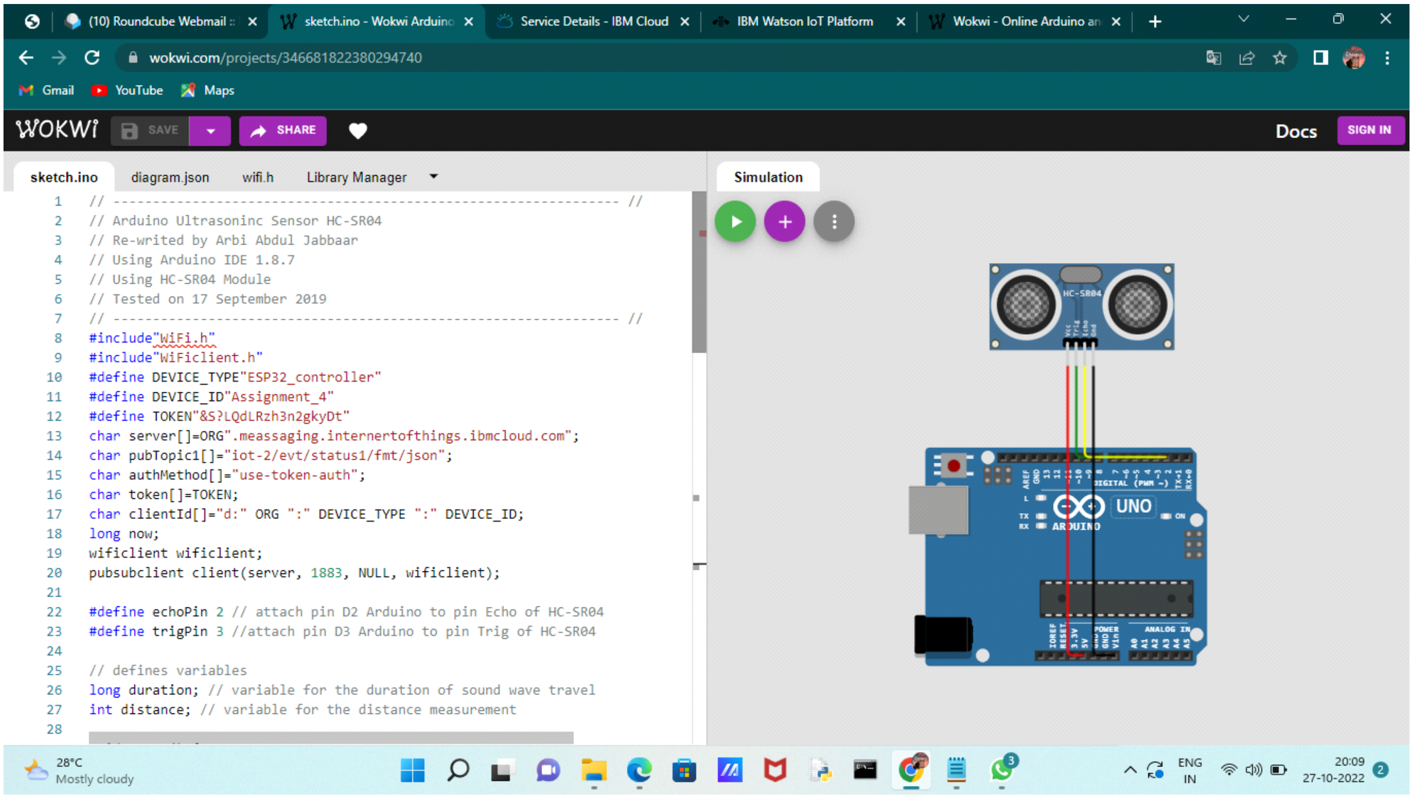
Assignment:4

Write code and connections in wokwi for ultrasonic sensor.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 image of ibm cloud

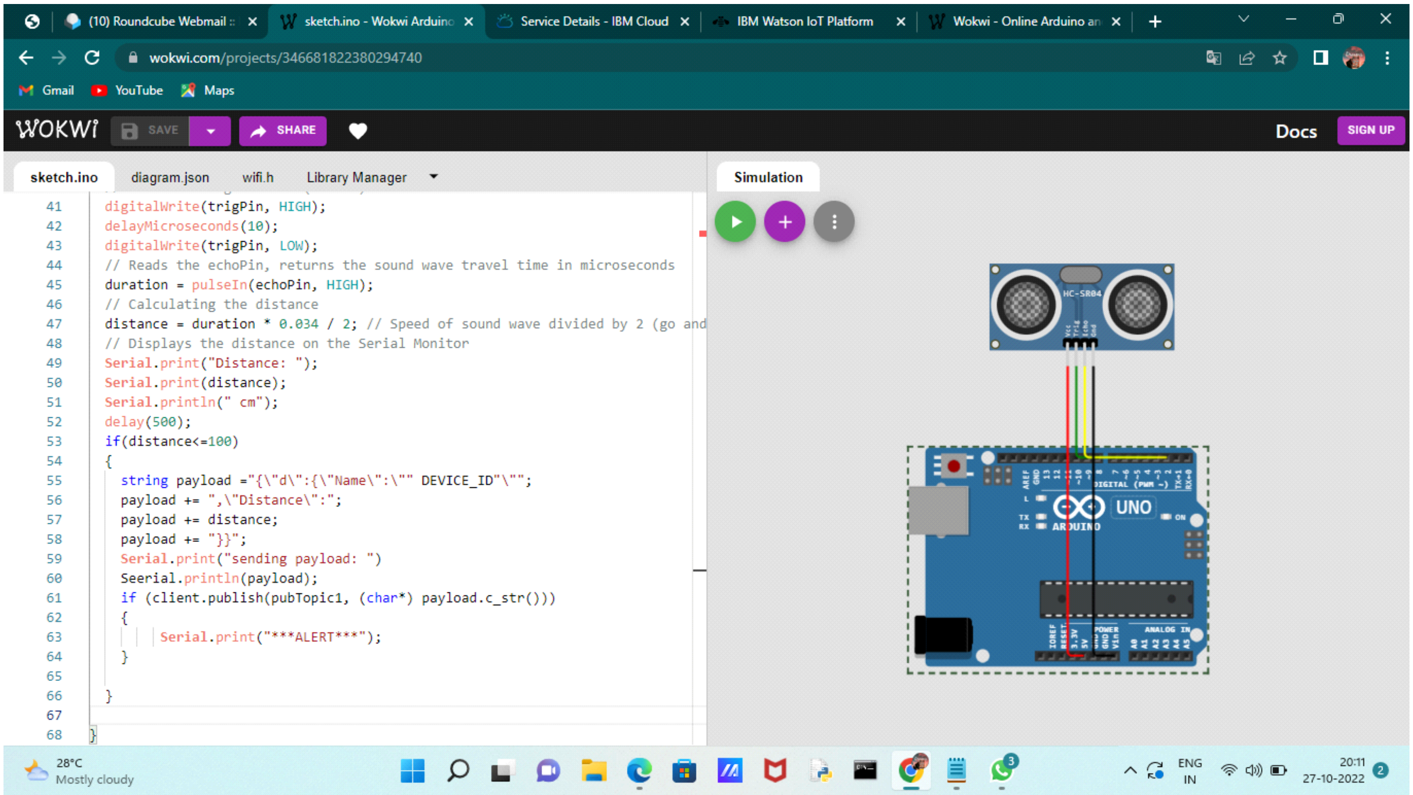
Code:

```
#include"WiFi.h"
#include"WiFiclient.h"
#define DEVICE_TYPE"ESP32_controller"
#define DEVICE_ID"Assignment_4"
#define TOKEN"&S?LQdLRzh3n2gkyDt"
char server[]=ORG".meassaging.internertofthings.ibmcloud.com";
char pubTopic1[]="iot-2/evt/status1/fmt/json";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientId[]="d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
long now;
wificlient wificlient;
pubsubclient client(server, 1883, NULL, wificlient);

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
```

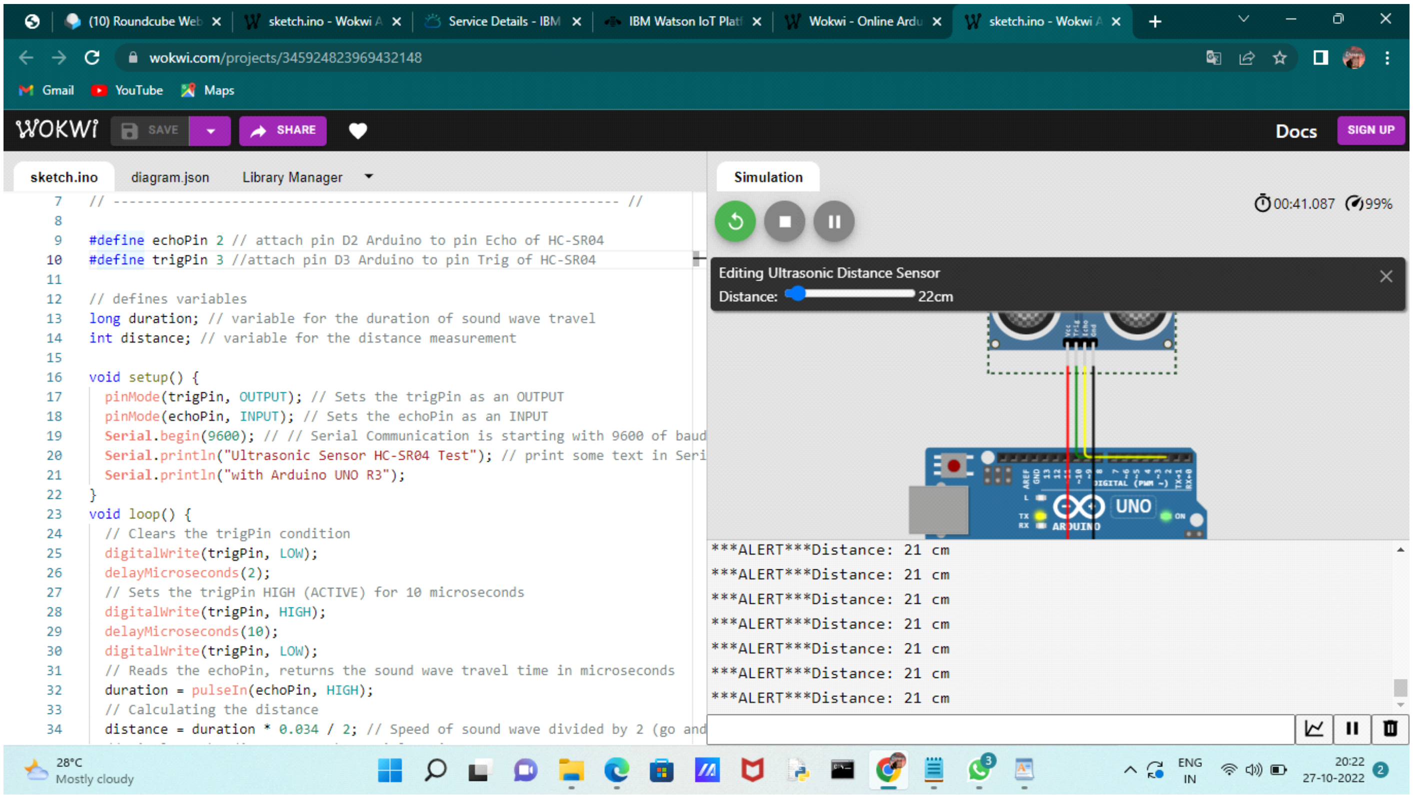
```
int distance; // variable for the distance measurement

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
}
void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
  if(distance<=100)
  {
    string payload ="{\"d\":{\"Name\":\"" DEVICE_ID"\"";
    payload += ",\"Distance\":";
    payload += distance;
    payload += "}}";
    Serial.print("sending payload: ")
    Seerial.println(payload);
    if (client.publish(pubTopic1, (char*) payload.c_str()))
    {
          Serial.print("***ALERT***");
    }

  }

}
```



```
// ----------------------------------------------------------------- //
// Arduino Ultrasoninc Sensor HC-SR04
// Re-writed by Arbi Abdul Jabbaar
// Using Arduino IDE 1.8.7
// Using HC-SR04 Module
// Tested on 17 September 2019
// ----------------------------------------------------------------- //
#include"WiFi.h"
#include"WiFiclient.h"
#define DEVICE_TYPE"ESP32_controller"
#define DEVICE_ID"Assignment_4"
#define TOKEN"&S?LQdLRzh3n2gkyDt"
char server[]=ORG".meassaging.internertofthings.ibmcloud.com";
char pubTopic1[]="iot-2/evt/status1/fmt/json";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientId[]="d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
long now;
wificlient wificlient;
pubsubclient client(server, 1883, NULL, wificlient);

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement

```



```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
  if(distance<=100)
  {
    string payload ="{\"d\":{\"Name\":\"" DEVICE_ID"\"";
    payload += ",\"Distance\":";
    payload += distance;
    payload += "}}";
    Serial.print("sending payload: ")
    Seerial.println(payload);
    if (client.publish(pubTopic1, (char*) payload.c_str()))
    {
        Serial.print("***ALERT***");
    }

  }

}
```

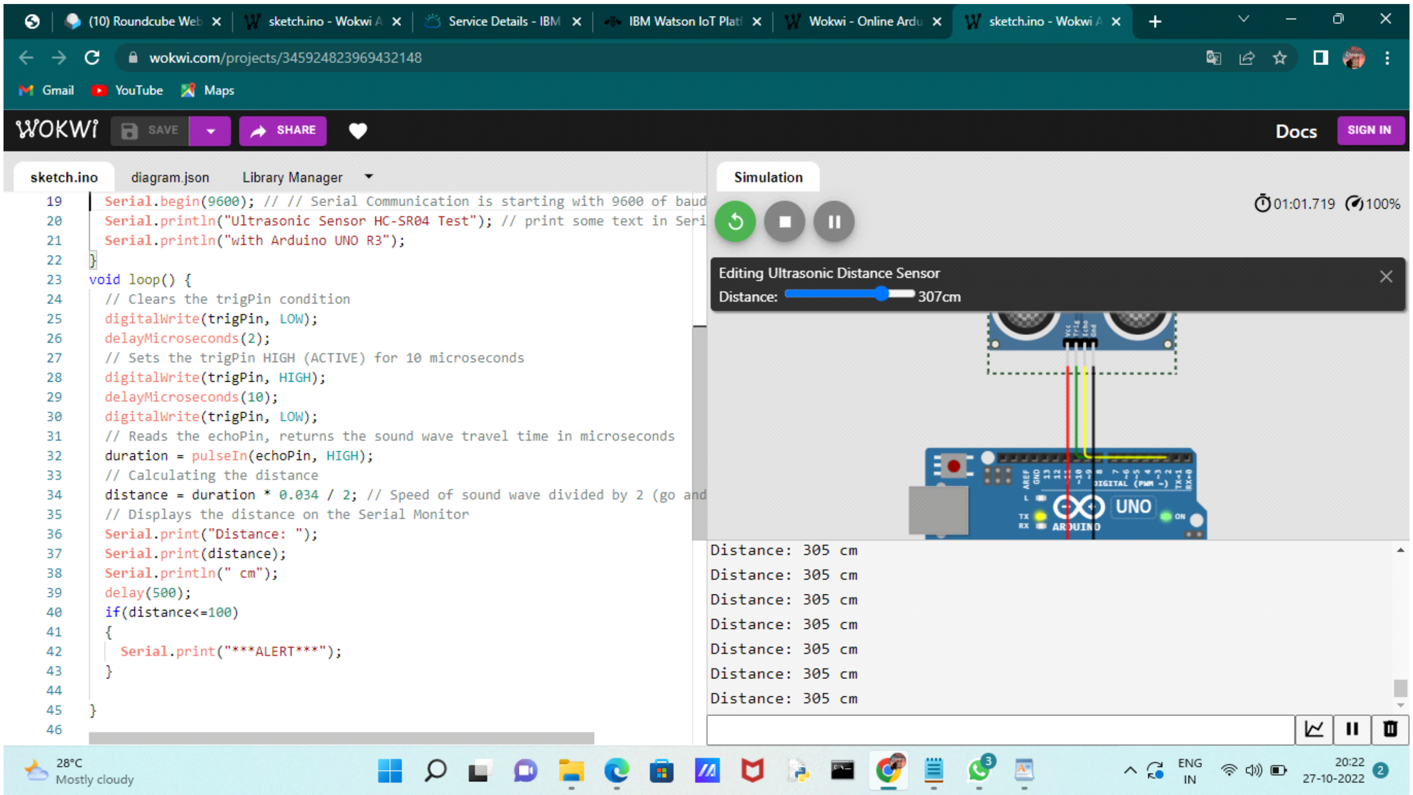


```
// ---------------------------------------------------------------- //

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baud
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Seri
  Serial.println("with Arduino UNO R3");
}
void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and
```

Editing Ultrasonic Distance Sensor

Distance: 22cm

```
***ALERT***Distance: 21 cm
***ALERT***Distance: 21 cm
***ALERT***Distance: 21 cm
***ALERT***Distance: 21 cm
***ALERT***Distance: 21 cm
***ALERT***Distance: 21 cm
***ALERT***Distance: 21 cm
```



```
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baud
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Seri
  Serial.println("with Arduino UNO R3");
}
void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
  if(distance<=100)
  {
    Serial.print("***ALERT***");
  }

}

```

Editing Ultrasonic Distance Sensor

Distance: 307cm

```
Distance: 305 cm
Distance: 305 cm
Distance: 305 cm
Distance: 305 cm
Distance: 305 cm
Distance: 305 cm
Distance: 305 cm
```

OUTPUT:

```
Distance: 305 cm

Distance: 40 cm

***ALERT***Distance: 40 cm
```