ASSIGNMENT-4

Assignment Date	13 November 2022
Student Name	**Mr. VINOTH**
Student Roll Number	**513419106044**
Maximum Marks	

QUESTION-1:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an “alert” to the IBM cloud and display in the device recent events.

Upload document with wokwi share link and images of IBM cloud.

SOLUTION:

Wokwi link: https://wokwi.com/projects/348216764594651730

CODE:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "6q4xt1"

#define DEVICE_TYPE "BlackSquid"

#define DEVICE_ID "12345"

#define TOKEN "w+U5*9o*h3W0I@A-tt"

#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);

  pinMode(echopin, INPUT);

  wifiConnect();

  mqttConnect();

}


void loop() {

  bool isNearby = dist < 100;

  digitalWrite(led, isNearby);


  publishData();

  delay(500);


  if (!client.loop()) {

    mqttConnect();
  }
}

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(client.subscribe(topic));
    Serial.println("WORKING");
  } else {
    Serial.println("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);
```

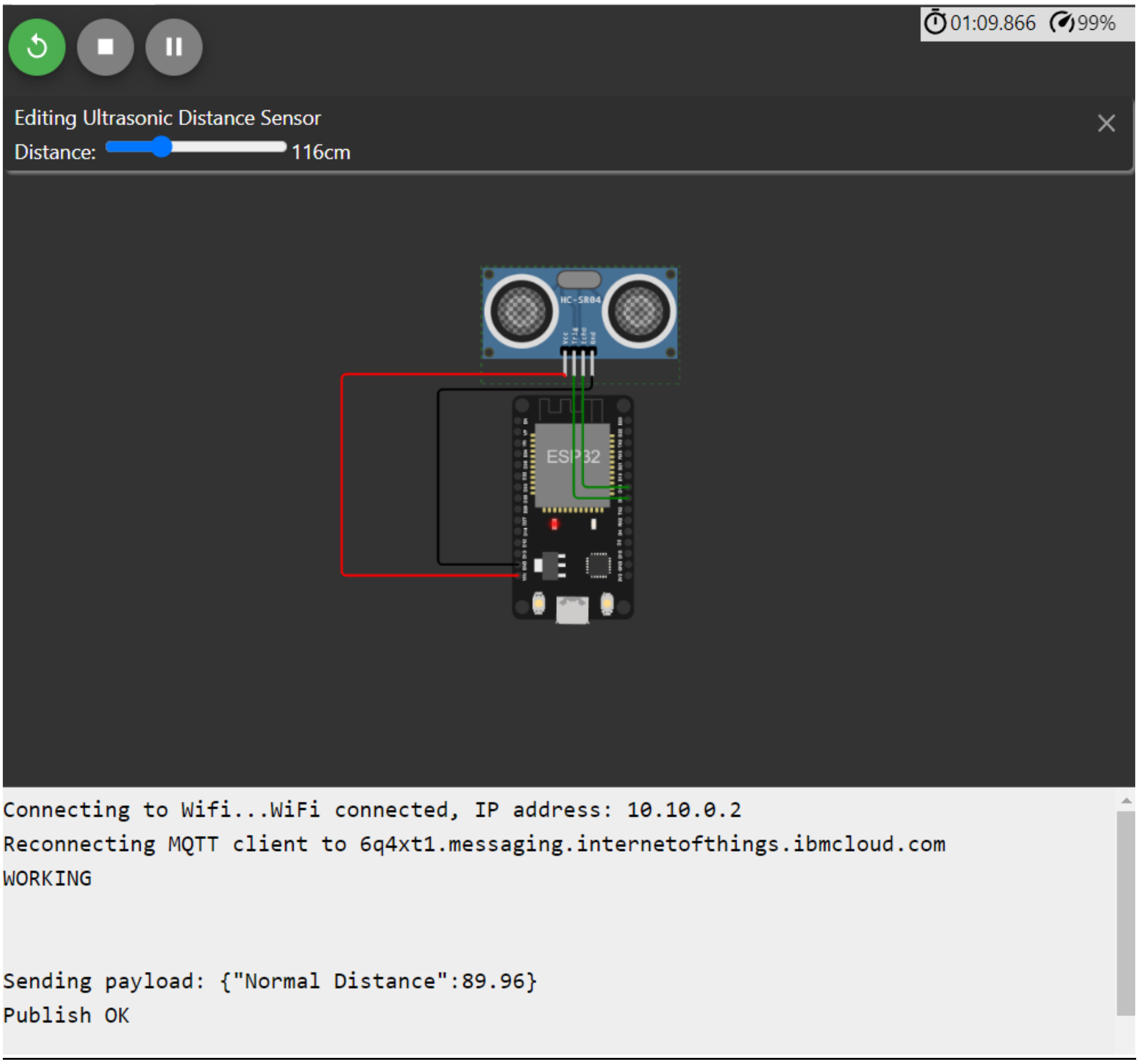
```
    if(client.publish(publishTopic, (char*) payload.c_str())) {
     Serial.println("Warning crosses 110cm -- it automaticaly of the loop");
     digitalWrite(led,HIGH);
   }else {
     Serial.println("Publish FAILED");
   }

  }



  }
  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"){
    Serial.println(data3);
    digitalWrite(led,HIGH);
  }
  data3="";
}
```

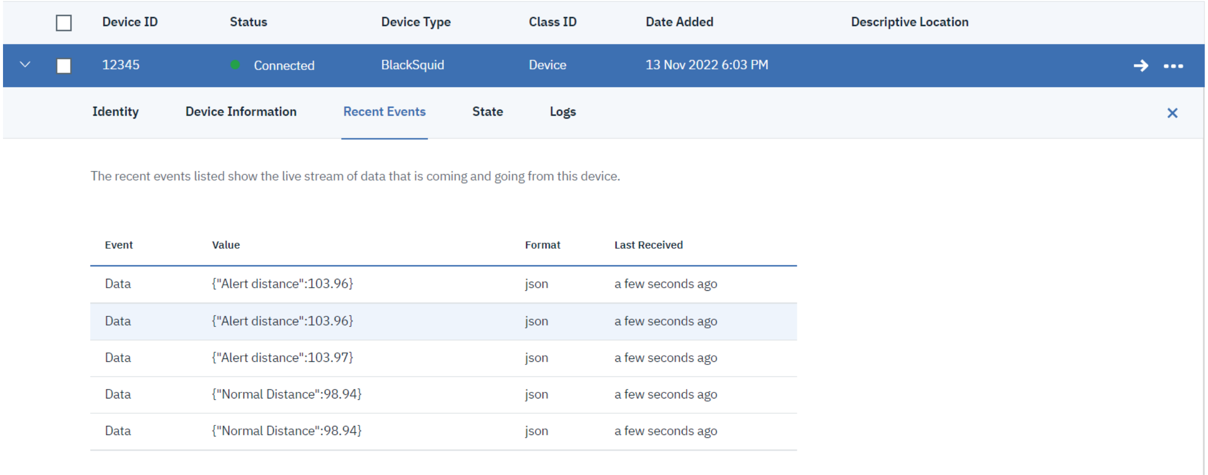
DIAGRAM:



```
Connecting to Wifi...WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to 6q4xt1.messaging.internetofthings.ibmcloud.com
WORKING


Sending payload: {"Normal Distance":89.96}
Publish OK
```

GETTING OUTPUT IN IBM WATSON



□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	12345	● Connected	BlackSquid	Device	13 Nov 2022 6:03 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Alert distance":103.96}	json	a few seconds ago
Data	{"Alert distance":103.96}	json	a few seconds ago
Data	{"Alert distance":103.97}	json	a few seconds ago
Data	{"Normal Distance":98.94}	json	a few seconds ago
Data	{"Normal Distance":98.94}	json	a few seconds ago