ASSIGNMENT 4

Date	04 November 2022
Team ID	PNT2022TMID38250
Student Name	DIVYA E
Student Roll.no	412319205010

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

CODE:

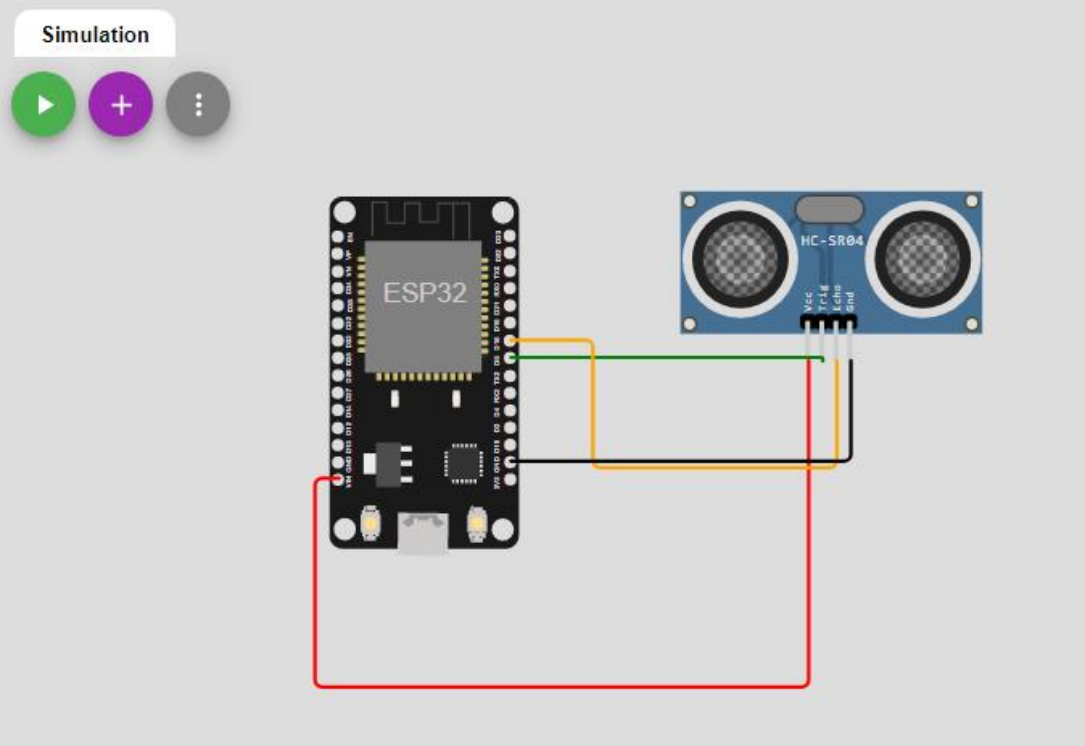
```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "6daage"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "29072001"
#define TOKEN "KreRD?fN18p&dx-rJr"
#define speed 0.034
String data;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/command/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;
```

```
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
//digitalWrite(trigpin,LOW);
float dur=pulseIn(echopin,HIGH);
float dist=(dur*0.0343)/2;
Serial.println("Distance");
Serial.println(dist,dur);
  publishData(dist,dur);
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void publishData(float dist,float dur)
{

  String obj;
  //float dur;
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  dur=pulseIn(echopin,HIGH);
  dist=dur*speed/2;

  if(dist<100){
    //digitalWrite(LED, HIGH);
    Serial.println("object is near");
    obj="Near";
```

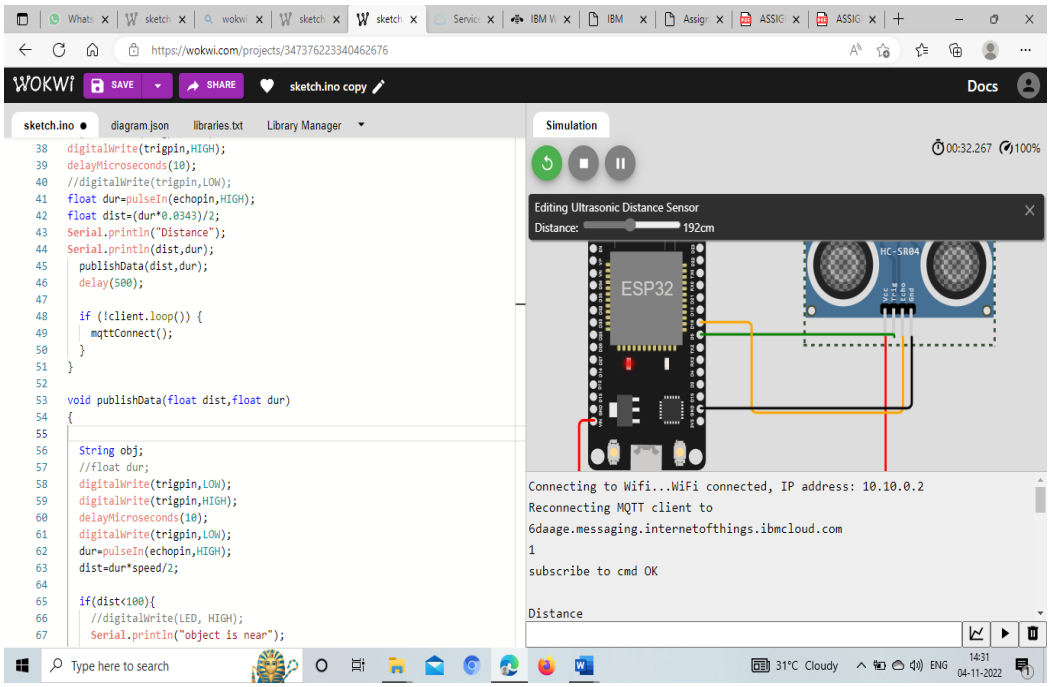
```
  }else{
    //digitalWrite(LED, HIGH);
   Serial.println("object is far");
   obj="far";
  }
  //DynamicJsonDocument doc(1024);
  String payload = "{\"Distance\":";
  payload += dist;
  payload += "," "\"Duration\":";
  payload += dur;
  payload += "}";

  delay(3000);
  Serial.print(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");
  }
}
void mqttConnect() {
  if (!client.connected()) {
   Serial.print("Reconnecting MQTT client to "); Serial.println(server);
   while (!client.connect(clientId, authMethod, token)) {
    Serial.print(".");
    delay(1000);
   }
   initManagedDevice();
   Serial.println();
  }
}
void initManagedDevice() {
  if (client.subscribe(topic)) {
    Serial.println(client.subscribe(topic));
   Serial.println("subscribe to cmd OK");
  } else {
   Serial.println("subscribe to cmd FAILED");
  }
}
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 callback(char* subscribetopic, byte* payload, unsigned int
payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data += (char)payload[i];
  }
    Serial.println("data: "+ data);
data="";
}
```

PICTURES:



OUTPUT:



WOKKI LINK:

[sketch.ino copy - Wokwi Arduino and ESP32 Simulator]

CLOUD PICTURE: