# ASSIGNMENT 4

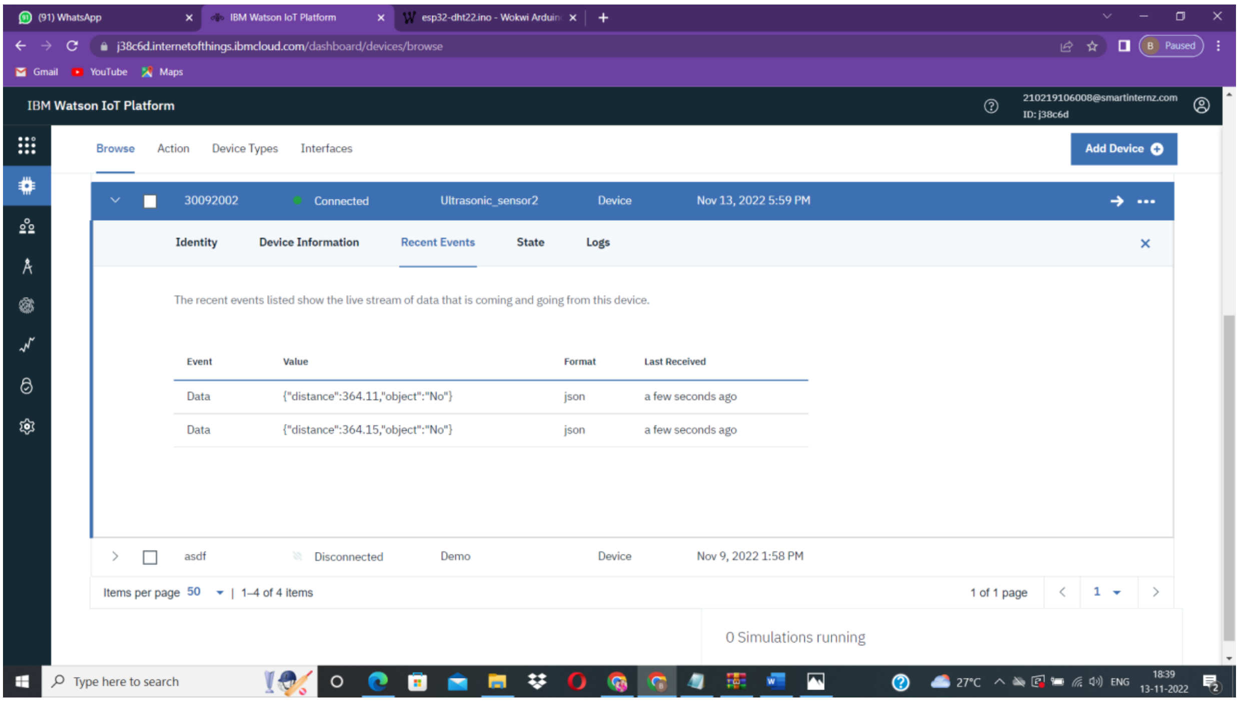
QUESTION 1:
Write code and connections in work for ultrasonic sensor. Whenever distance is less than 100 cm send “alert” to IBM cloud and display in device recent events.

## CODE:

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


void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

//-------credentials of IBM Accounts------

#define ORG "j38c6d"//IBM ORGANITION ID
#define DEVICE_TYPE "Ultrasonic_sensor2"//Device type mentioned in ibm
watson IOT Platform
#define DEVICE_ID "30092002"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "Y*u5EP43OwTC!P_D*s"     //Token
String data3;
float dist;
//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of
event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT
command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential

int trig = 5;
int echo = 18;
void setup()
{
Serial.begin(115200);
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
delay(10);
wificonnect();
mqttconnect();
}
```

```
void loop()// Recursive Function
{

 digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  float dur = pulseIn(echo,HIGH);
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distancein cm");
  Serial.println(dist);


PublishData(dist);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
void PublishData(float dist)
{
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String object;
  if (dist <100)
  {

    Serial.println("object is near");
    object = "Near";
  }
  else
  {

    Serial.println("no object found");
    object = "No";
  }

  String payload = "{\"distance\":";
  payload += dist;
  payload += "," "\"object\":\"";
  payload += object;
  payload += "\"}";


  Serial.print("Sending payload: ");
  Serial.println(payload);




  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the
cloud then it will print publish ok in Serial monitor or else it will
print publish failed
  } else {
    Serial.println("Publish failed");
  }

}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd 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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  //   Serial.println("data: "+ data3);
//   if(data3=="Near")
//   {
// Serial.println(data3);
// digitalWrite(LED,HIGH);
//    }

//    else
//    {
// Serial.println(data3);
// digitalWrite(LED,LOW);

//    }
data3="";
}
```

# OUTPUT:



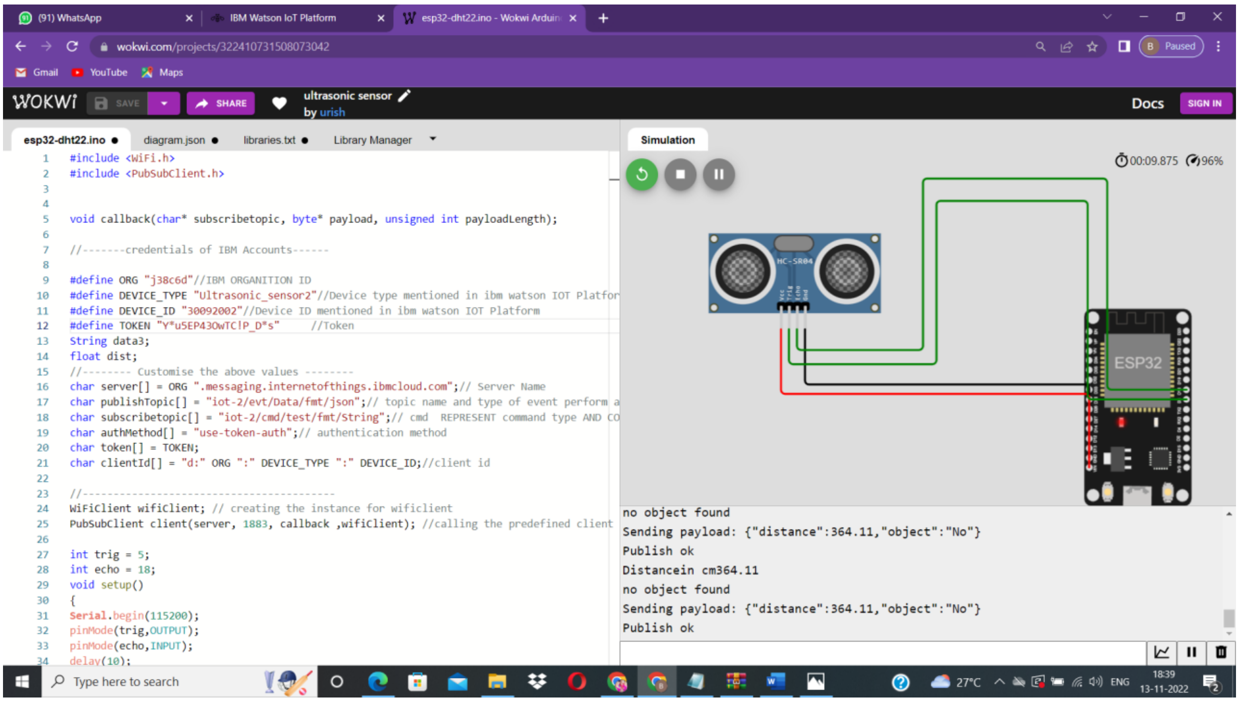
| | 30092002 | Connected | Ultrasonic_sensor2 | Device | Nov 13, 2022 5:59 PM |
|---|---|---|---|---|---|

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

| Event | Value | Format | Last Received |
|---|---|---|---|
| Data | {"distance":364.11,"object":"No"} | json | a few seconds ago |
| Data | {"distance":364.15,"object":"No"} | json | a few seconds ago |

| | asdf | Disconnected | Demo | Device | Nov 9, 2022 1:58 PM |
|---|---|---|---|---|---|

Data send to the IBM cloud device when the objects is less than 100 cm



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


void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "j38c6d"//IBM ORGANITION ID
#define DEVICE_TYPE "Ultrasonic_sensor2"//Device type mentioned in ibm watson IOT Platfor
#define DEVICE_ID "30092002"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "Y*u5EP43OwTC!P_D*s"     //Token
String data3;
float dist;
//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform a
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND CO
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client

int trig = 5;
int echo = 18;
void setup()
{
Serial.begin(115200);
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
delay(10);
```

```
no object found
Sending payload: {"distance":364.11,"object":"No"}
Publish ok
Distancein cm364.11
no object found
Sending payload: {"distance":364.11,"object":"No"}
Publish ok
```