Assignment -4

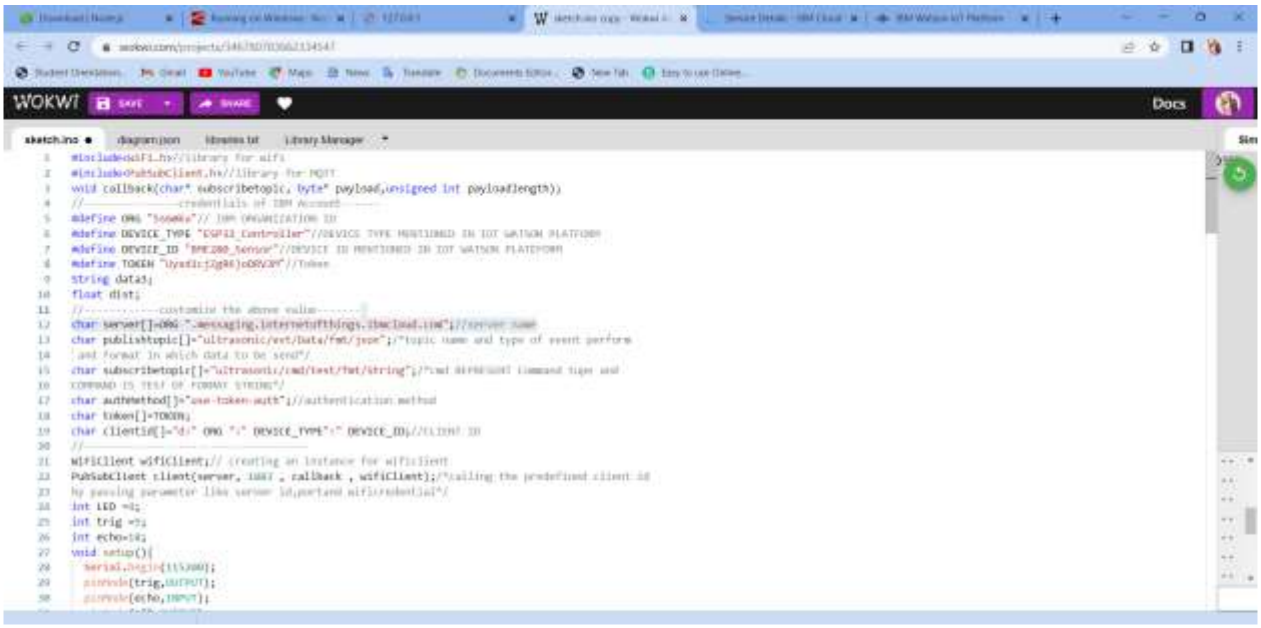
Wokwi Programming

Assignment Date	29 October 2022
Student Name	K.NATHIYA
Student Roll Number	912419104021
Maximum Marks	2 Marks

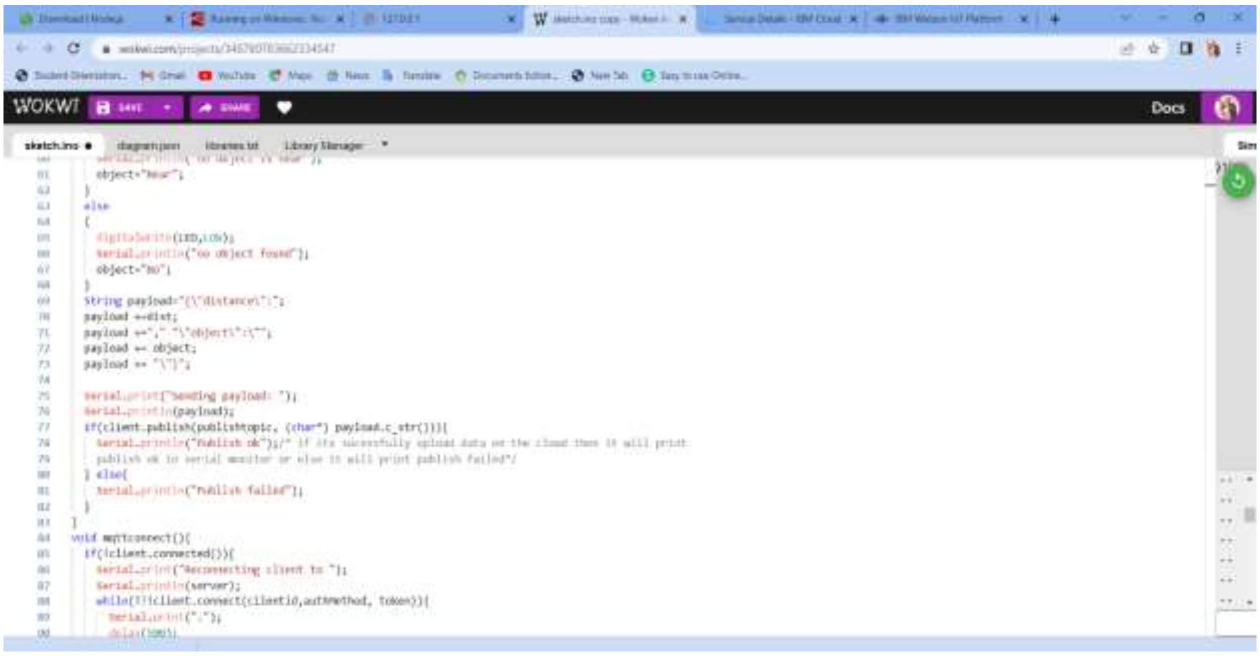
Question-1:

Write code and connections in wokwi for ultrasonic sensor.
Whenever distance is less than 100 cms send "alert" to ibm cloud and display in devicerecent events.
Upload document with wokwi share link and images of ibm cloud.

Solution:
Coding:

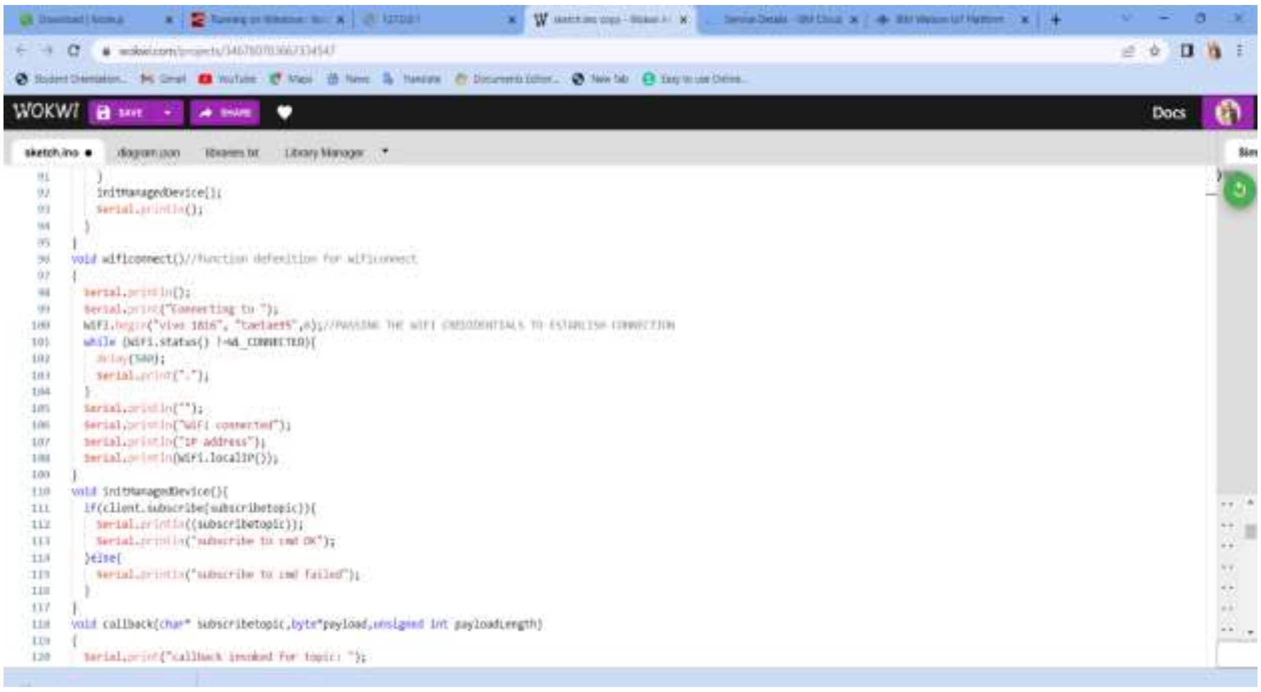


```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload,unsigned int payloadlength));
//-------------credentials of IBM Account------
#define ORG "[illegible]"// IBM ORGANIZATION ID
#define DEVICE_TYPE "ESP32_Controller"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "BME280_Sensor"//DEVICE ID MENTIONED IN IOT WATSON PLATFORM
#define TOKEN "[illegible]"//Token
String data3;
float dist;
//-----------customize the above value-------
char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name
char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and type of event perform
 and format in which data to be send*/
char subscribetopic[]="ultrasonic/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 an instance for wificlient
PubSubClient client(server, 1883 , callback , wifiClient);/*calling the predefined client id
by passing parameter like server id,portand wificredential*/
int LED =4;
int trig =5;
int echo=18;
void setup(){
  Serial.begin(115200);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
```

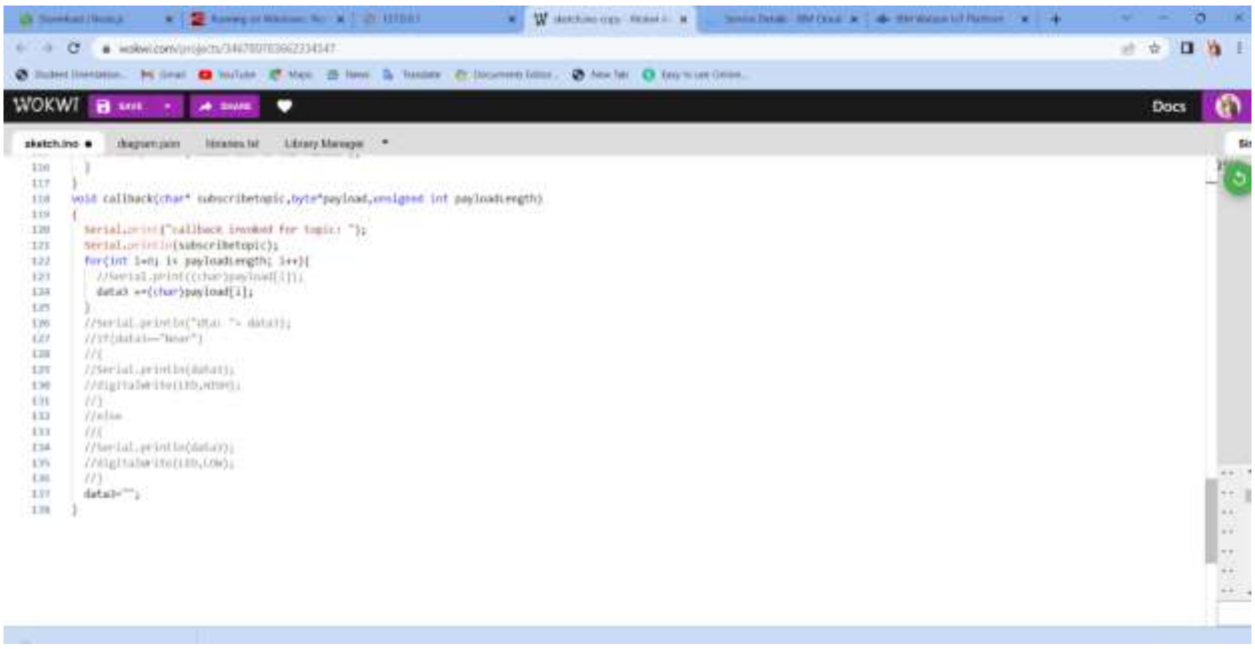


```
    object="Near";
  }
  else
  {
    digitalWrite(LED,LOW);
    Serial.println("no object found");
    object="No";
  }
  String payload="{\"distance\":";
  payload +=dist;
  payload +=","  "\"object\":\"";
  payload += object;
  payload += "\"}";

  Serial.print("Sending payload: ");
  Serial.println(payload);
  if(client.publish(publishtopic, (char*) payload.c_str())){
    Serial.println("Publish ok");/* if its 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);
```

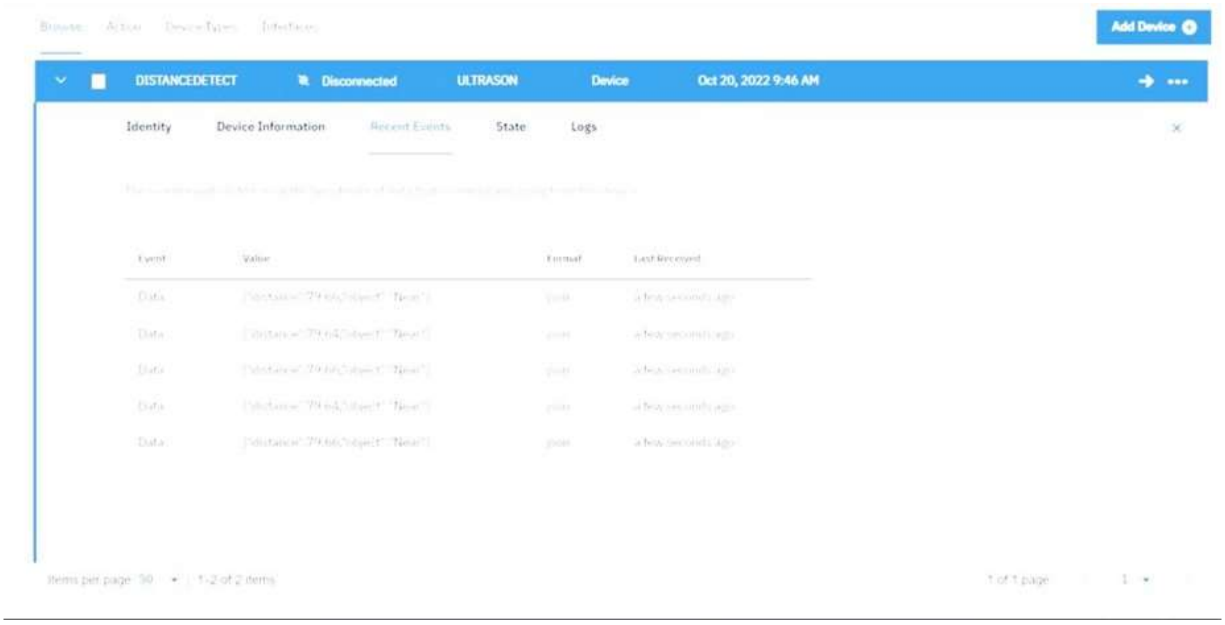


```
    }
    initManagedDevice();
    Serial.println();
  }
}
void wificonnect()//function defenition for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("vivo 1816", "taetaet$",6);//PASSING THE WIFI CREDIDENTIALS TO ESTABLISH CONNECTION
  while (WiFi.status() !=WL_CONNECTED){
    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: ");
```



```
  }
}
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="";
}
```

DATA SENT TO IBM CLOUD ON NO OBJECT DETECTED



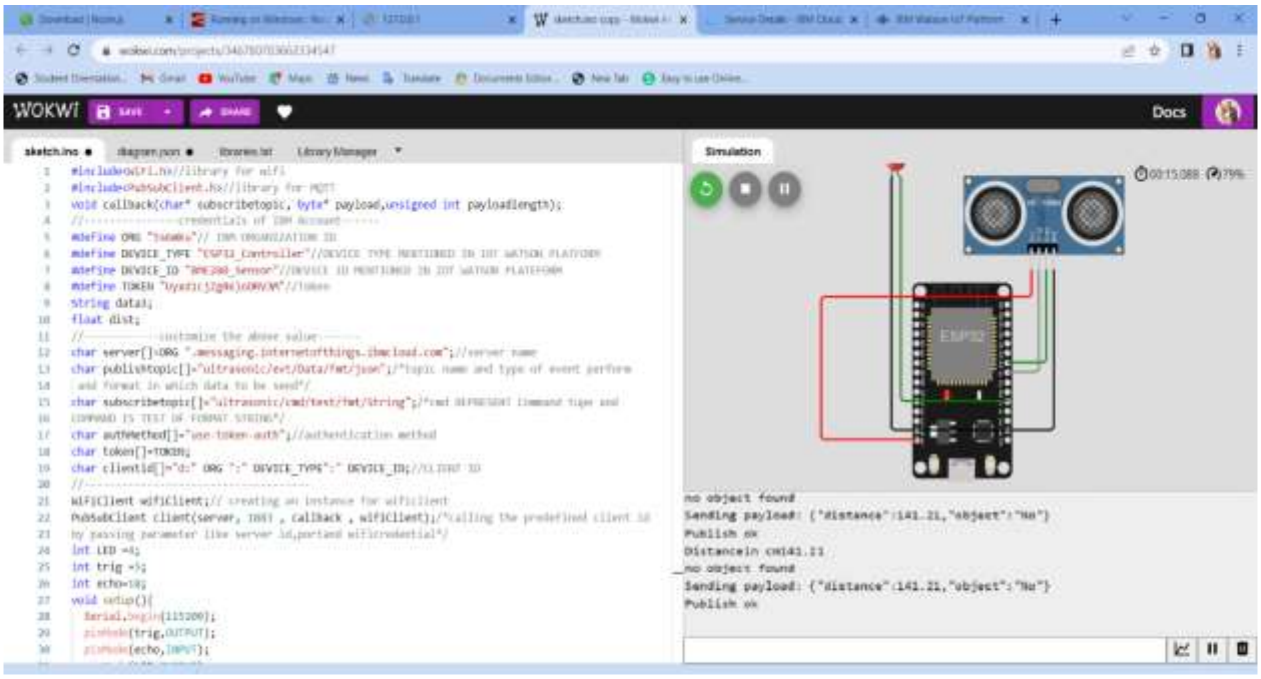
DISTANCEDETECT Disconnected ULTRASON Device Oct 20, 2022 9:46 AM

Identity Device Information Recent Events State Logs

Event	Value	Format	Last Received
Data	{"distance":79.66,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.64,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.66,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.64,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.66,"object":"Near"}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items | 1 of 1 page

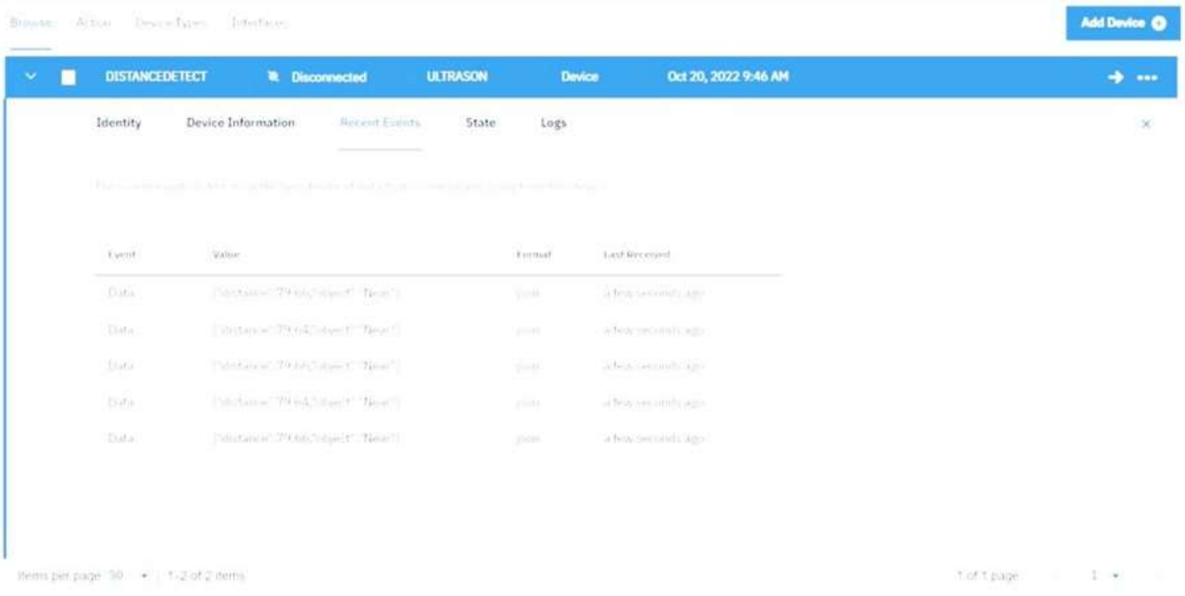
WHEN NO OBJECT DETECTED BY ULTRASONIC DETECTOR



```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload,unsigned int payloadlength);
//--------------credentials of IBM Account------
#define ORG "[illegible]"// IBM ORGANIZATION ID
#define DEVICE_TYPE "ESP32_Controller"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "BME280_Sensor"//DEVICE ID MENTIONED IN IOT WATSON PLATEFORM
#define TOKEN "[illegible]"//Token
String data3;
float dist;
//----------customize the above value------
char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name
char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and type of event perform
 and format in which data to be send*/
char subscribetopic[]="ultrasonic/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 an instance for wificlient
PubSubClient client(server, 1883 , callback , wifiClient);/*calling the predefined client id
by passing parameter like server id,portand wificredential*/
int LED =4;
int trig =5;
int echo=18;
void setup(){
  Serial.begin(115200);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
```

```
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
Distancein cm141.21
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
```

DATA SENT TO IBM CLOUD ON OBJECT BEING DETECTED

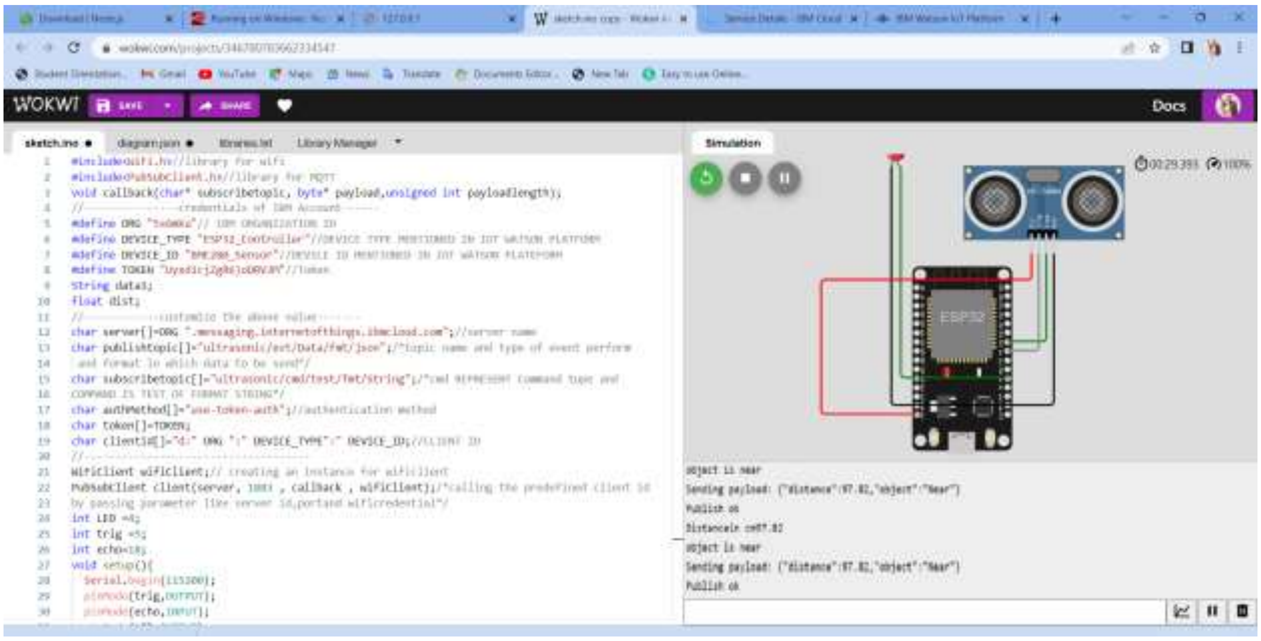


DISTANCEDETECT	Disconnected	ULTRASON	Device	Oct 20, 2022 9:46 AM

Identity | Device Information | Recent Events | State | Logs

Event	Value	Format	Last Received
Data	{"distance":79.66,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.64,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.66,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.64,"object":"Near"}	json	a few seconds ago
Data	{"distance":79.66,"object":"Near"}	json	a few seconds ago

Items per page: 50 | 1-2 of 2 items | 1 of 1 page



```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload,unsigned int payloadlength);
//-------------credentials of IBM Account------
#define ORG "[illegible]"// IBM ORGANISATION ID
#define DEVICE_TYPE "ESP32_Controller"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "BME280_Sensor"//DEVICE ID MENTIONED IN IOT WATSON PLATFORM
#define TOKEN "[illegible]"//Token
String data3;
float dist;
//-----------customize the above value------
char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name
char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and type of event perform
 and format in which data to be send*/
char subscribetopic[]="ultrasonic/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 an instance for wificlient
PubSubClient client(server, 1883 , callback , wifiClient);/*calling the predefined client id
by passing parameter like server id,portand wificredential*/
int LED =4;
int trig =5;
int echo=18;
void setup(){
  Serial.begin(115200);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
```

```
object is near
Sending payload: {"distance":87.82,"object":"Near"}
Publish ok
Distance in cm87.82
object is near
Sending payload: {"distance":87.82,"object":"Near"}
Publish ok
```

OUTPUT:

https://wokwi.com/projects/346780783662334547