Assignment -4

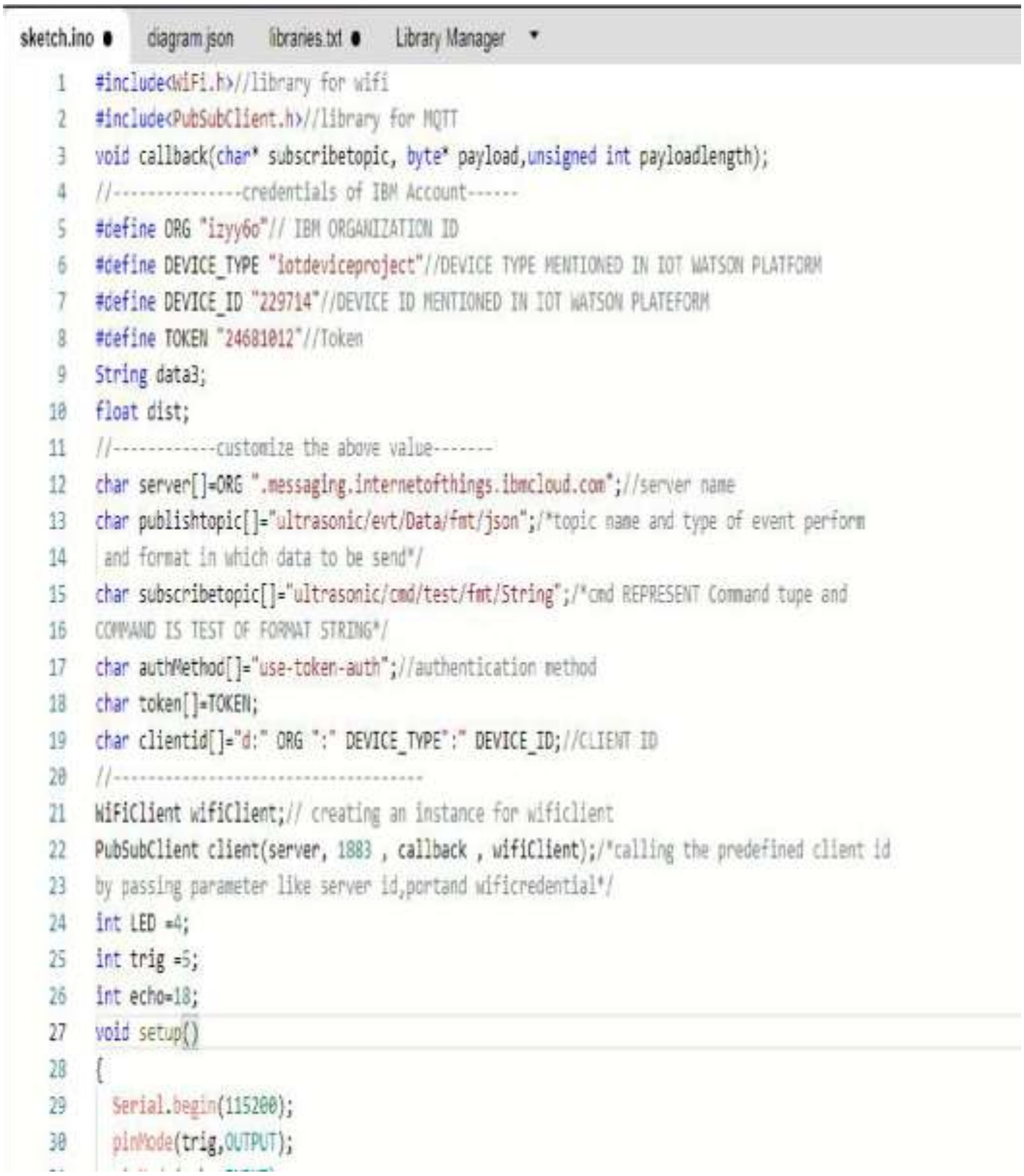
Python Programming

Assignment Date	31 October 2022
Student Name	PURNIMA R G
Student Roll Number	2031T304
Maximum Marks	2 Marks

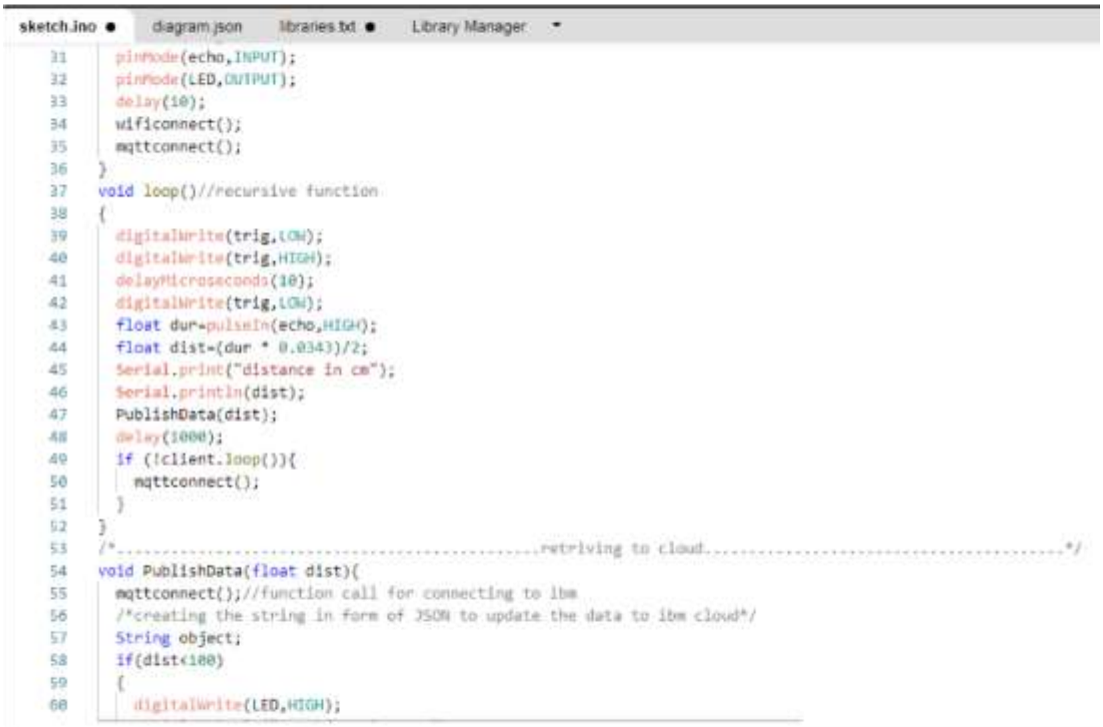
Question-1:

Write code and connections in wokwi for ultrasonic sensor.

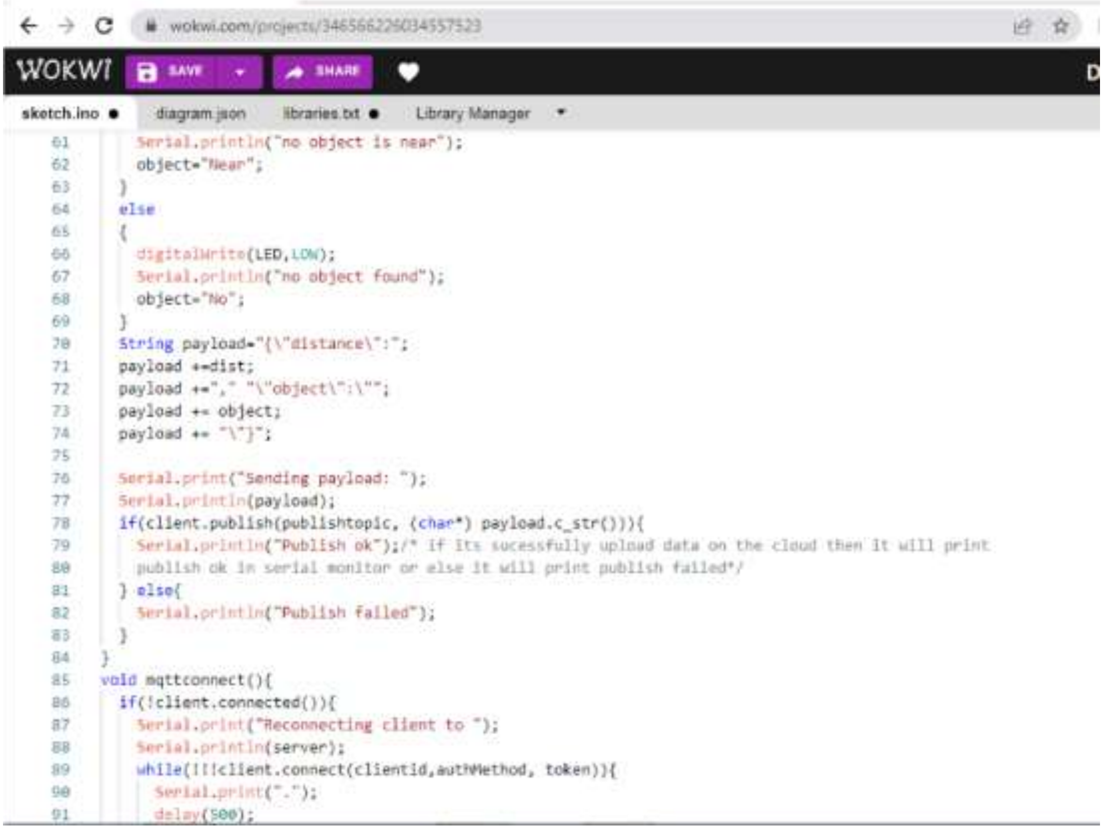
Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.



```
#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 "izyy6o"// IBM ORGANIZATION ID
#define DEVICE_TYPE "iotdeviceproject"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "229714"//DEVICE ID MENTIONED IN IOT WATSON PLATEFORM
#define TOKEN "24681012"//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 tupe 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);
```

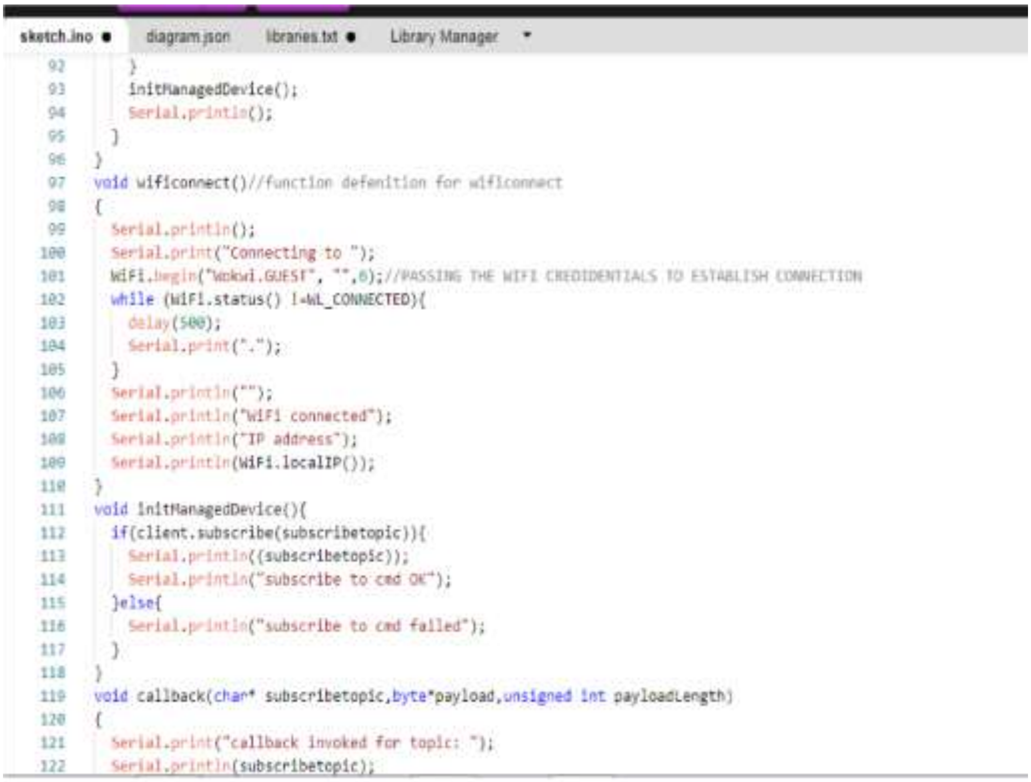


```
  pinMode(echo,INPUT);
  pinMode(LED,OUTPUT);
  delay(10);
  wificonnect();
  mqttconnect();
}
void loop()//recursive function
{
  digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  float dur=pulseIn(echo,HIGH);
  float dist=(dur * 0.0343)/2;
  Serial.print("distance in cm");
  Serial.println(dist);
  PublishData(dist);
  delay(1000);
  if (!client.loop()){
    mqttconnect();
  }
}
/*................................................retriving to cloud........................................*/
void PublishData(float dist){
  mqttconnect();//function call for connecting to ibm
  /*creating the string in form of JSON to update the data to ibm cloud*/
  String object;
  if(dist<100)
  {
    digitalWrite(LED,HIGH);
```

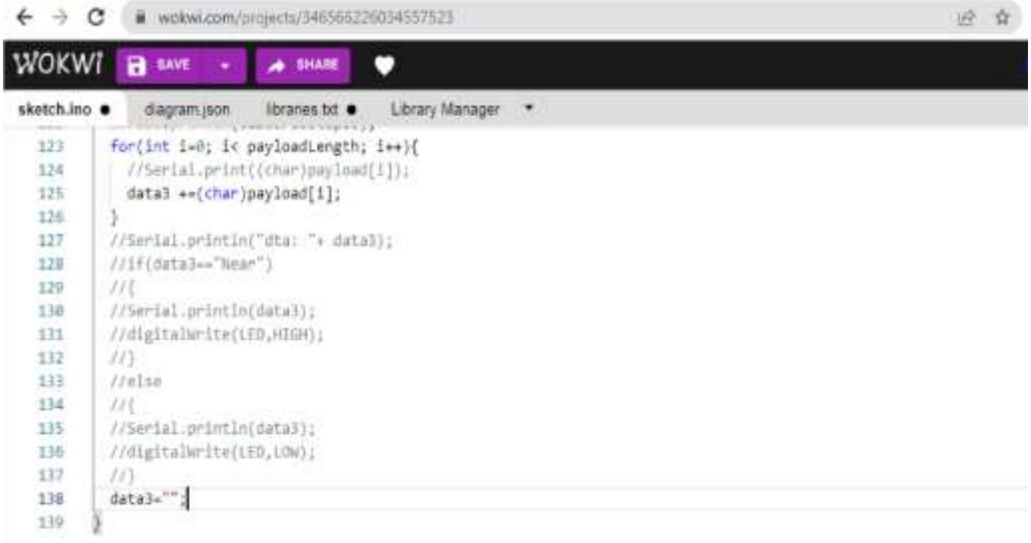


```
    Serial.println("no object is near");
    object="Near";
  }
  else
  {
    digitalWrite(LED,LOW);
    Serial.println("no object found");
    object="No";
  }
  String payload="{\"distance\":";
  payload +=dist;
  payload +="," "\"object\":\"";
  payload += object;
  payload += "\"}";

  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("Wokwi.GUEST", "",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: ");
  Serial.println(subscribetopic);
```

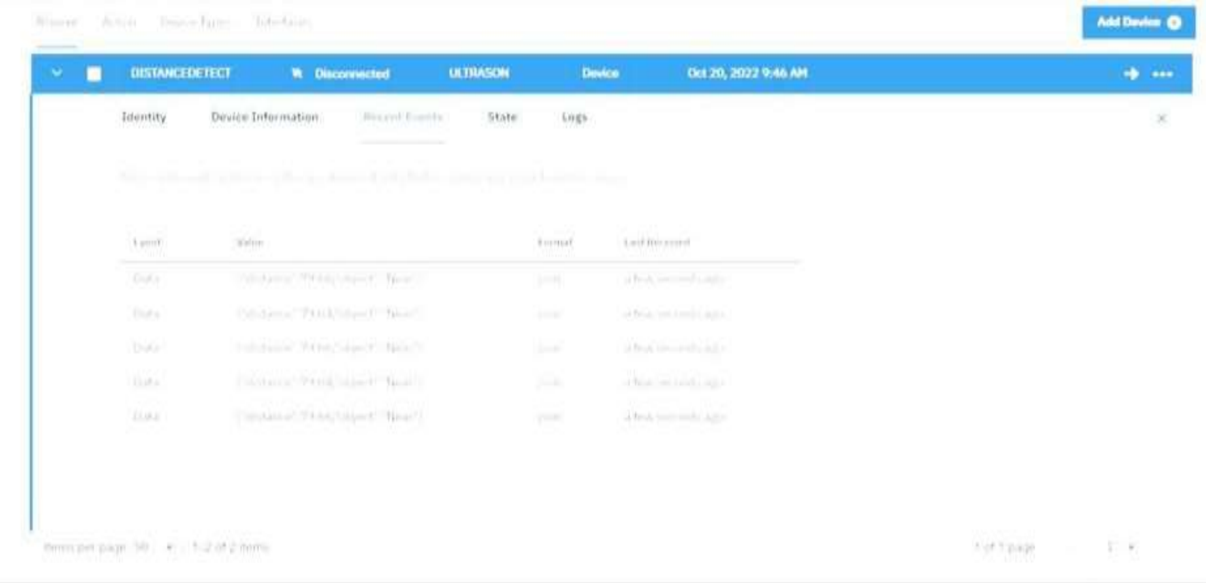


```
  for(int i=0; i< payloadLength; i++){
    //Serial.print((char)payload[i]);
    data3 +=(char)payload[i];
  }
  //Serial.println("dta: "+ data3);
  //if(data3=="Near")
  //{
  //Serial.println(data3);
  //digitalWrite(LED,HIGH);
  //}
  //else
  //{
  //Serial.println(data3);
  //digitalWrite(LED,LOW);
  //}
  data3="";
}
```

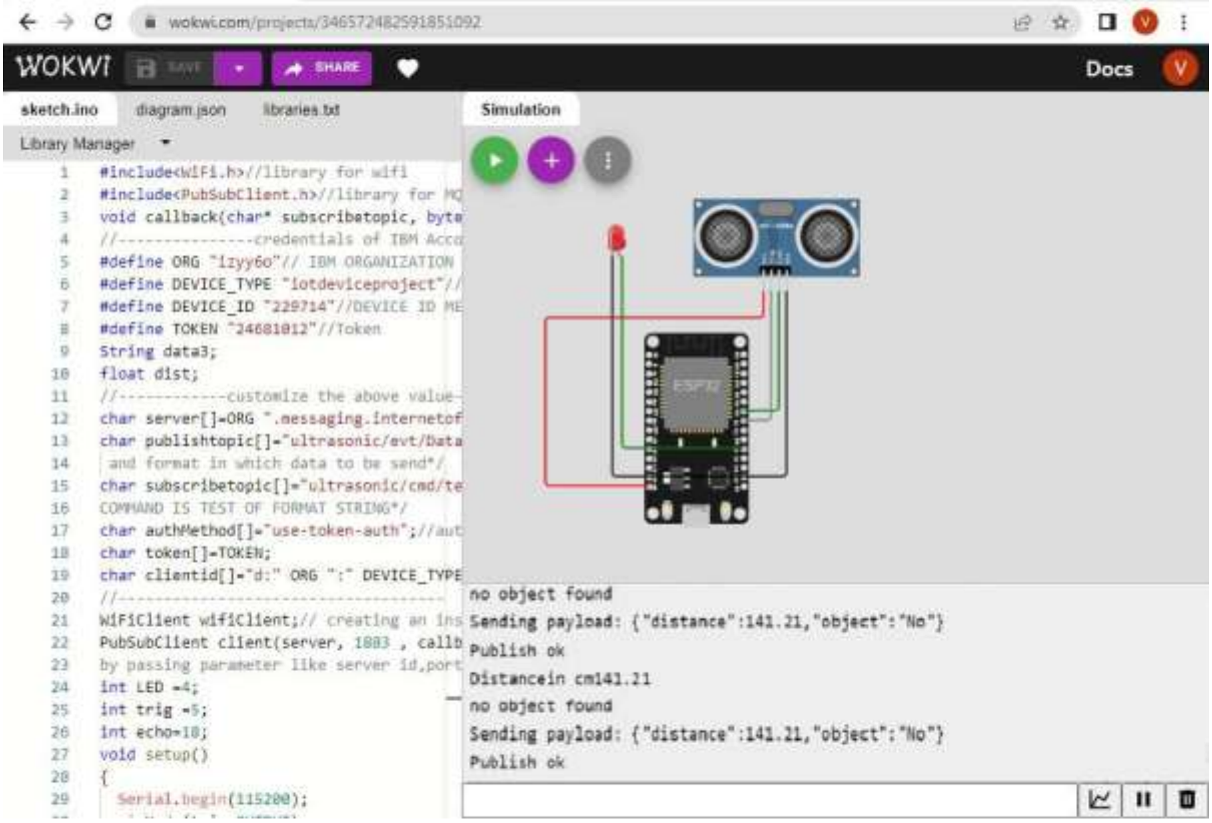
OUTPUT:

DATA IS SENT TO IBM CLOUD WHEN NO OBJECT IS DETECTED

When no object is detected



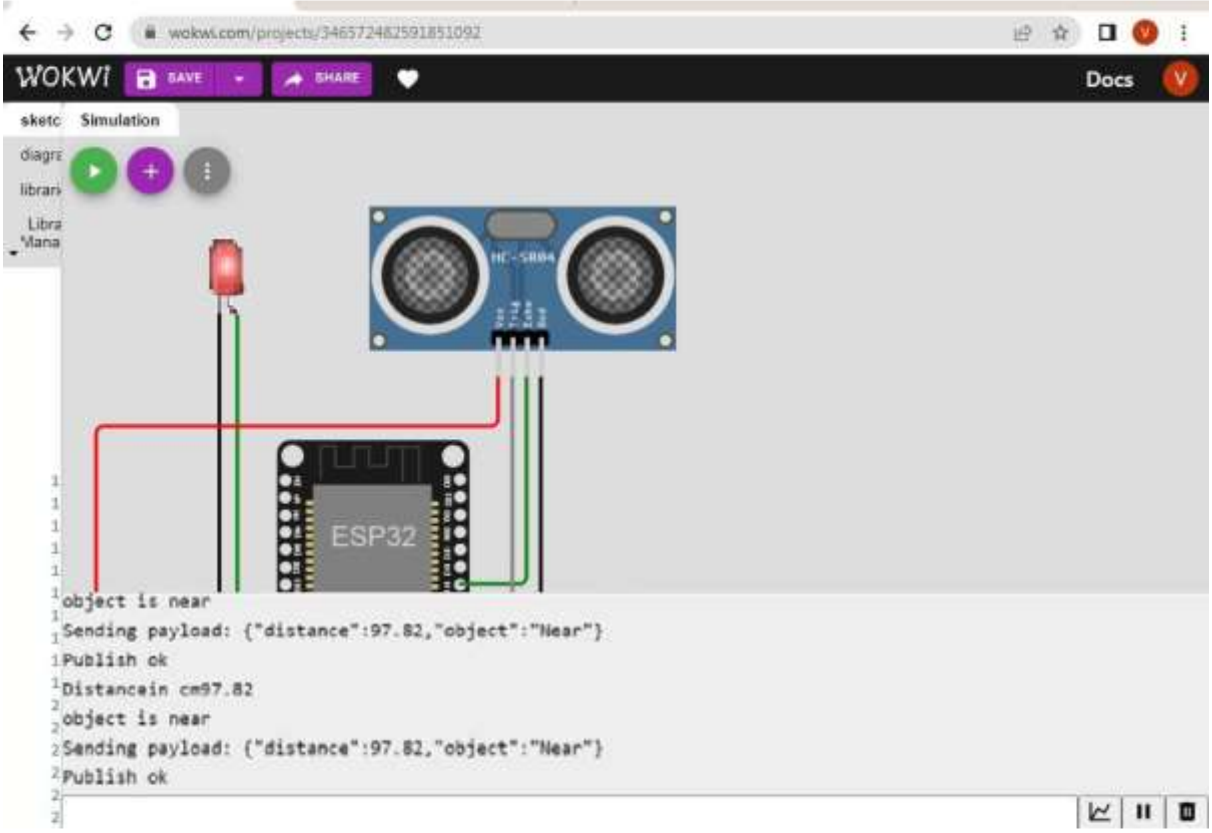
Event	Value	Format	Last Received
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago



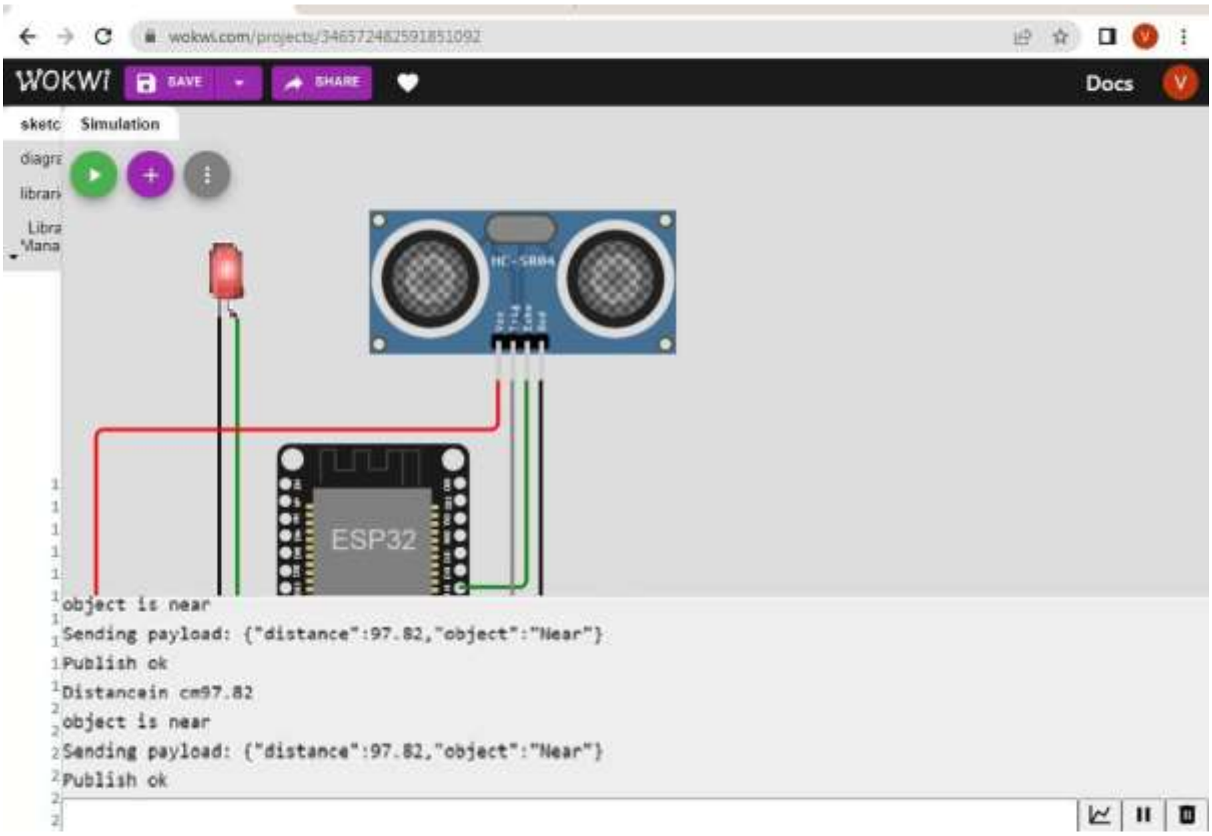
```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQ
void callback(char* subscribetopic, byte
//---------------credentials of IBM Acco
#define ORG "izyy6o"// IBM ORGANIZATION
#define DEVICE_TYPE "iotdeviceproject"//
#define DEVICE_ID "229714"//DEVICE ID ME
#define TOKEN "24681012"//Token
String data3;
float dist;
//------------customize the above value-
char server[]=ORG ".messaging.internetof
char publishtopic[]="ultrasonic/evt/Data
 and format in which data to be send*/
char subscribetopic[]="ultrasonic/cmd/te
COMMAND IS TEST OF FORMAT STRING*/
char authMethod[]="use-token-auth";//aut
char token[]=TOKEN;
char clientid[]="d:" ORG ":" DEVICE_TYPE
//-----------------------------------
WiFiClient wifiClient;// creating an ins
PubSubClient client(server, 1883 , callb
by passing parameter like server id,port
int LED =4;
int trig =5;
int echo=18;
void setup()
{
  Serial.begin(115200);
```

```
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 is sent to ibm cloud when object is detected



```
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
Distancein cm97.82
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
```



```
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
Distancein cm97.82
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
```