ASSIGNMENT - 4

DATE	17 OCTOBER 2022
TEAM ID	PNT2022TMID06972
NAME	SHOWMINI R
STUDENT ROLL NUMBER	GCTC1918L16
MAXIMUM MARKS	2 MARKS

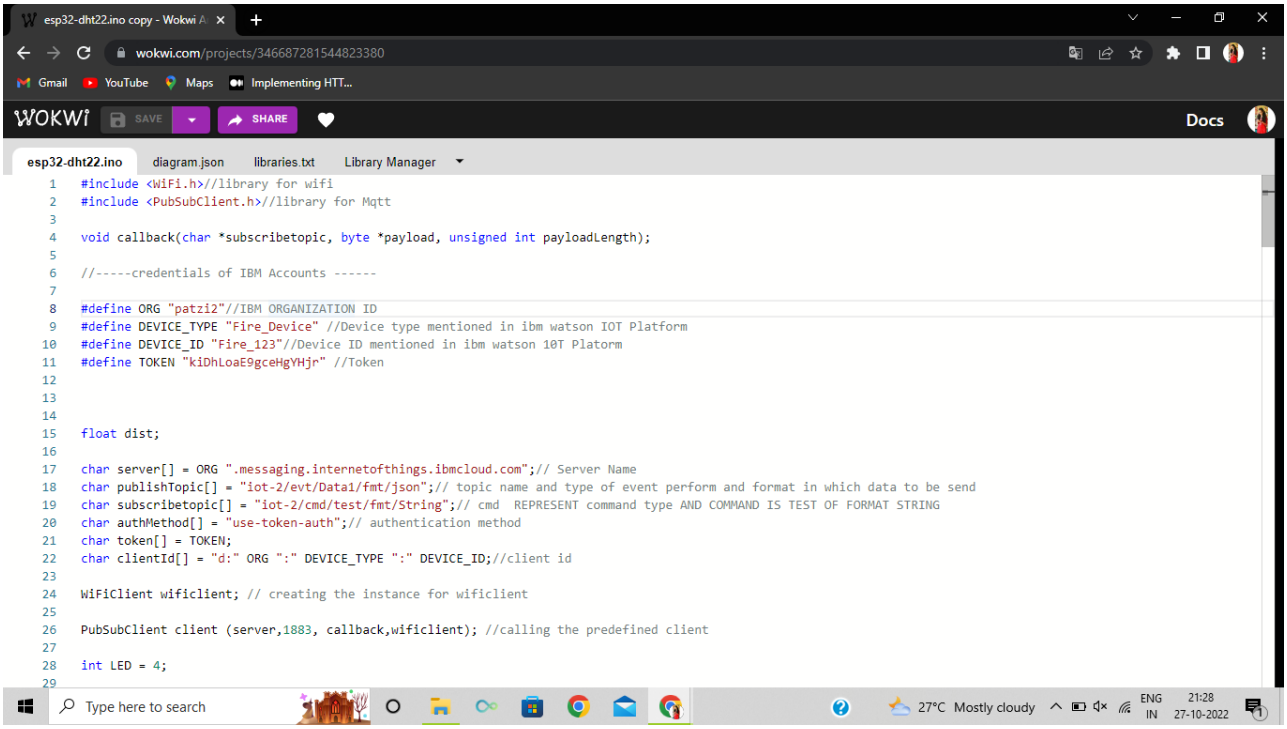
QUESTION:

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

WOKWI CODE AND IMPLEMENTATION LINK:

https://wokwi.com/projects/346688722332287572

CODE:



esp32-dht22.ino copy - Wokwi A

wokwi.com/projects/346687281544823380

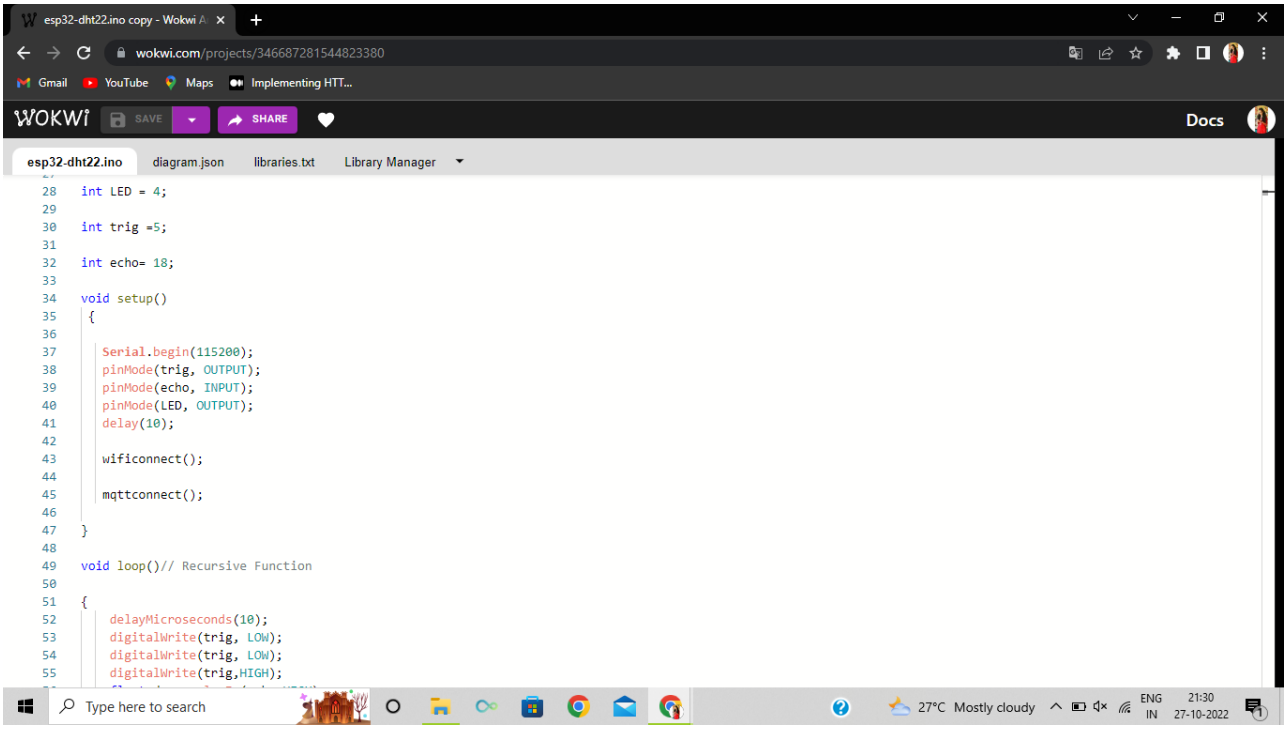
Gmail YouTube Maps Implementing HTT...

WOKWI SAVE SHARE Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for Mqtt

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

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

#define ORG "patzi2"//IBM ORGANIZATION ID
#define DEVICE_TYPE "Fire_Device" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Fire_123"//Device ID mentioned in ibm watson 10T Platorm
#define TOKEN "kiDhLoaE9gceHgYHjr" //Token



float dist;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data1/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

int LED = 4;
```

Type here to search 27°C Mostly cloudy ENG IN 21:28 27-10-2022



esp32-dht22.ino copy - Wokwi A

wokwi.com/projects/346687281544823380

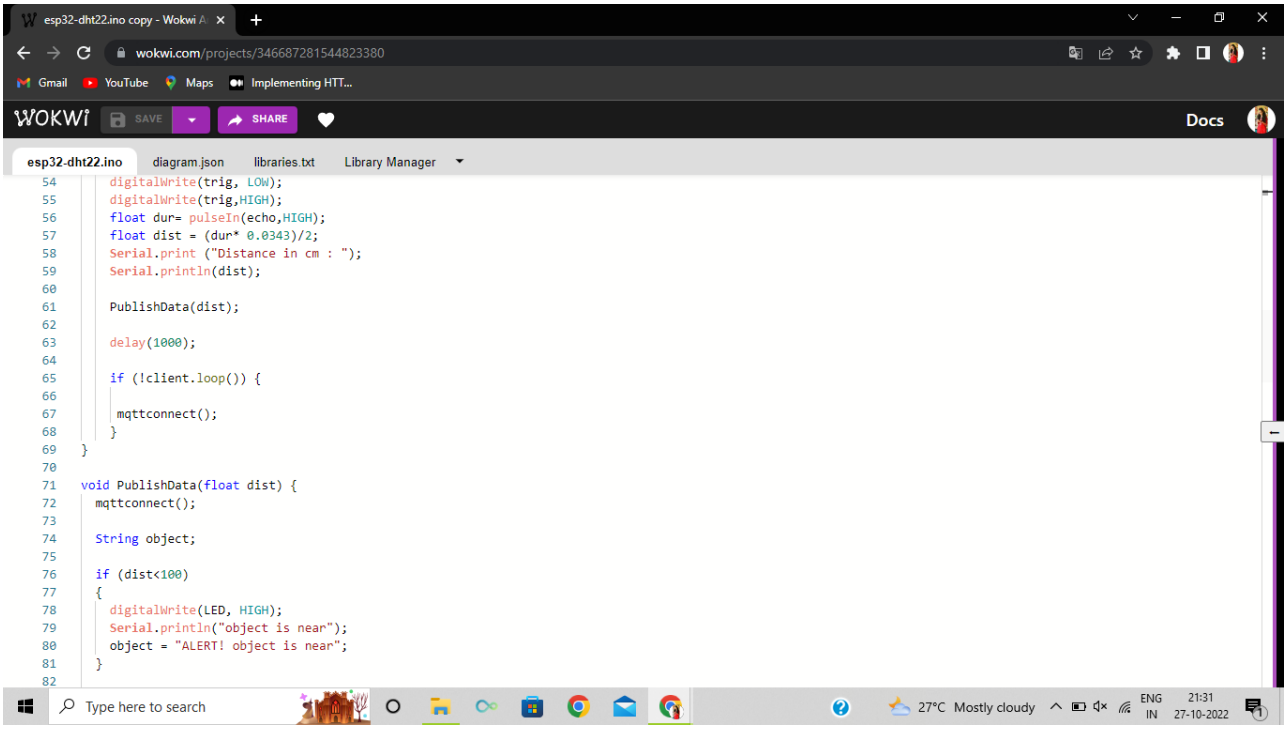
Gmail YouTube Maps Implementing HTT...

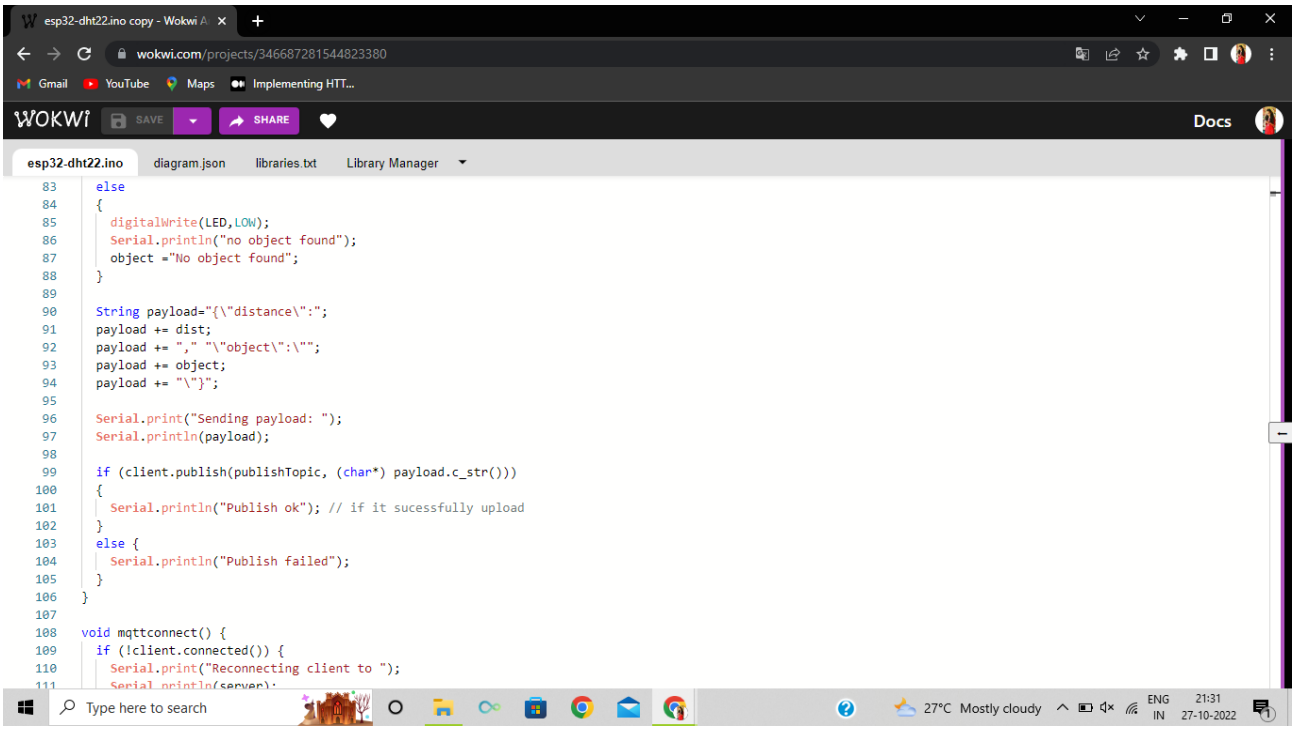
WOKWI SAVE SHARE Docs

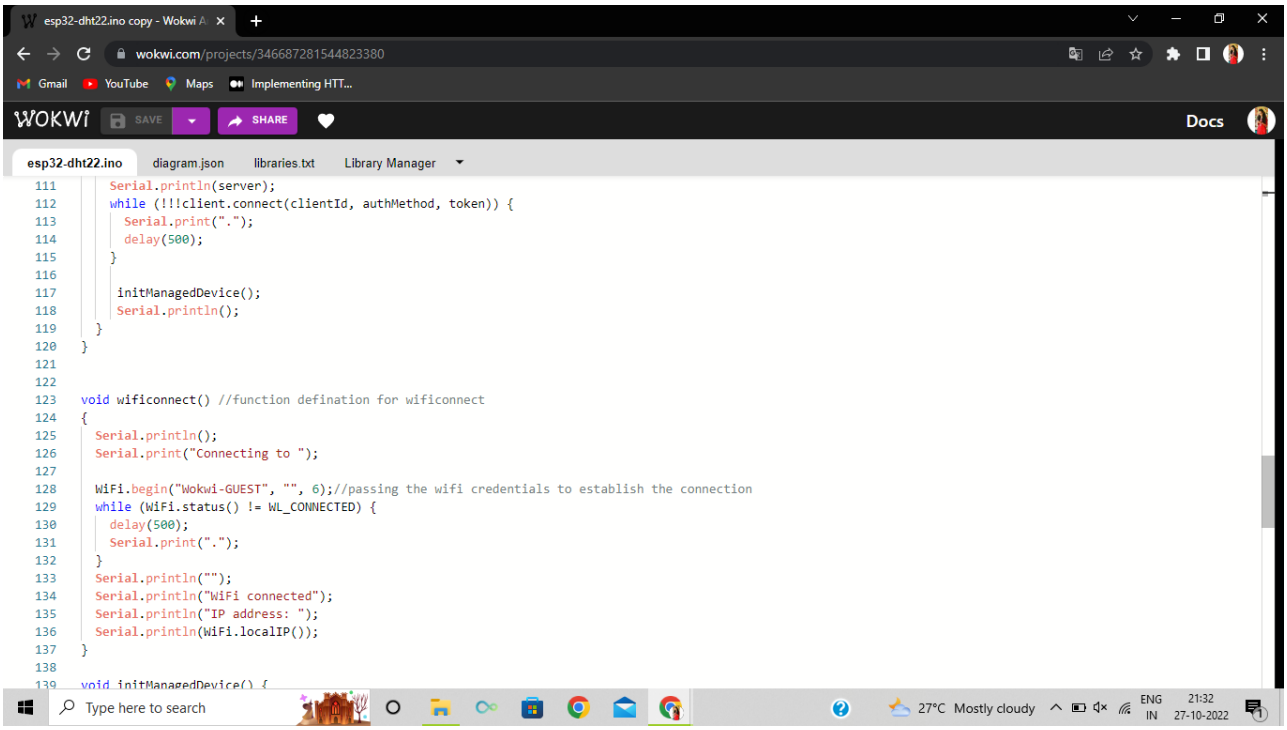
esp32-dht22.ino diagram.json libraries.txt Library Manager

```
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

{
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    digitalWrite(trig, LOW);
    digitalWrite(trig,HIGH);
```

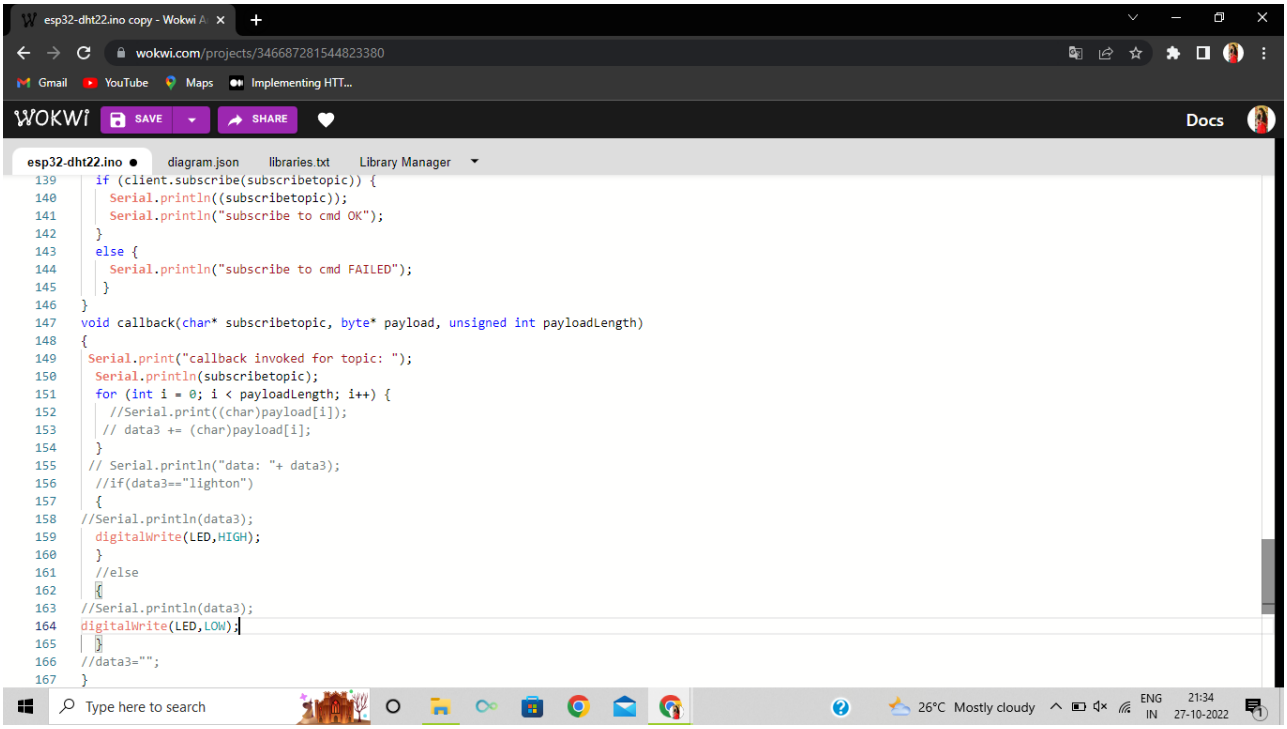
Type here to search 27°C Mostly cloudy ENG IN 21:30 27-10-2022


esp32-dht22.ino copy - Wokwi A
wokwi.com/projects/346687281544823380
Gmail
YouTube
Maps
Implementing HTT...
WOKWi
SAVE
SHARE
Docs
esp32-dht22.ino
diagram.json
libraries.txt
Library Manager
digitalWrite(trig, LOW);
digitalWrite(trig,HIGH);
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();
}
}
void PublishData(float dist) {
mqttconnect();
String object;
if (dist<100)
{
digitalWrite(LED, HIGH);
Serial.println("object is near");
object = "ALERT! object is near";
}
Type here to search
27°C Mostly cloudy
ENG
IN
21:31
27-10-2022


esp32-dht22.ino copy - Wokwi A
wokwi.com/projects/346687281544823380
Gmail
YouTube
Maps
Implementing HTT...
WOKWi
SAVE
SHARE
Docs
esp32-dht22.ino
diagram.json
libraries.txt
Library Manager
else
{
digitalWrite(LED,LOW);
Serial.println("no object found");
object ="No object found";
}
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
}
else {
Serial.println("Publish failed");
}
}
void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Type here to search
27°C Mostly cloudy
ENG
IN
21:31
27-10-2022



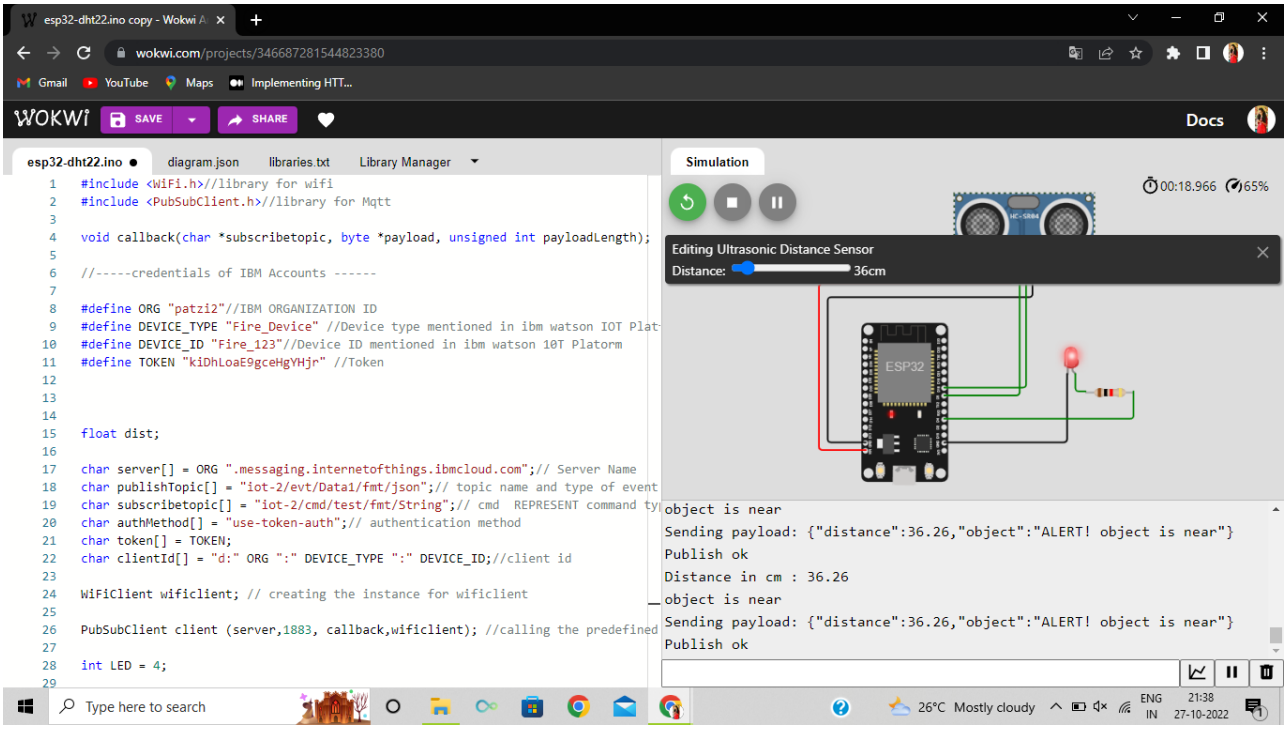
```
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

    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() {
```



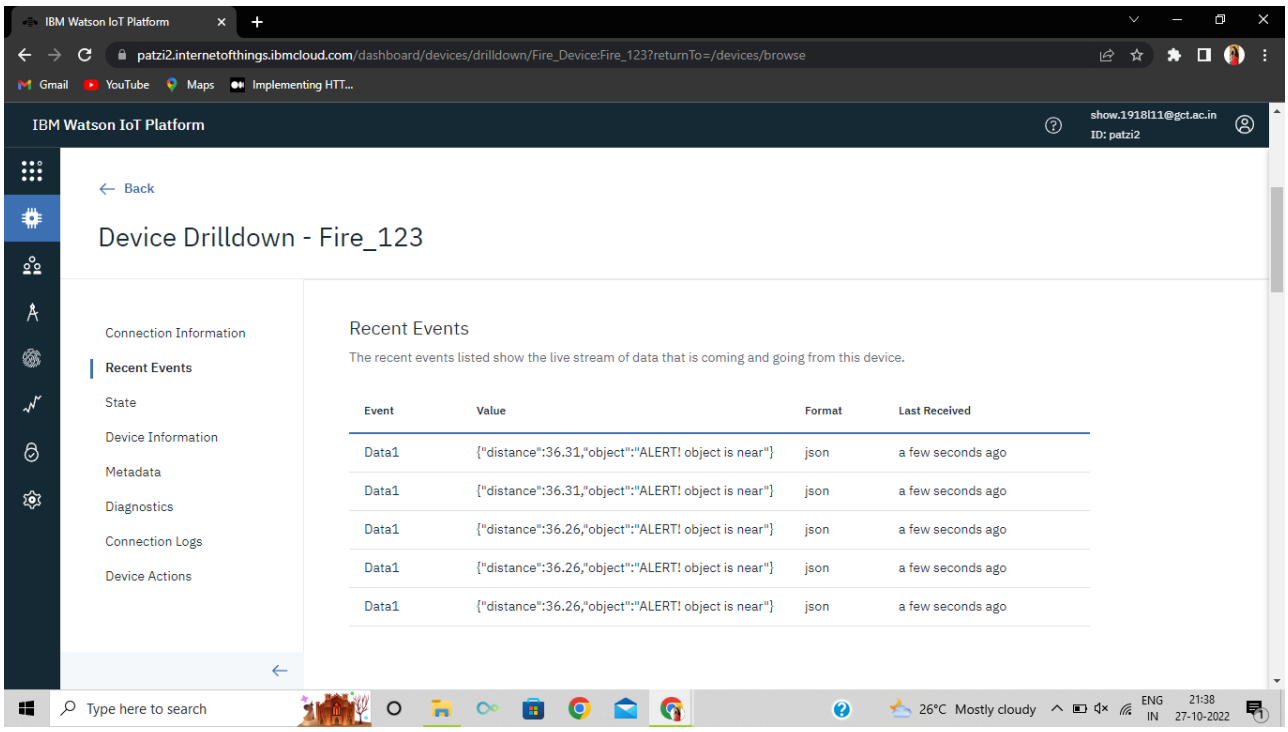
```
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  }
  else {
    Serial.println("subscribe to cmd FAILED");
    }
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
 Serial.print("callback invoked 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=="lighton")
  {
//Serial.println(data3);
  digitalWrite(LED,HIGH);
  }
  //else
  {
//Serial.println(data3);
digitalWrite(LED,LOW);
  }
  //data3="";
}
```

OUTPUT:

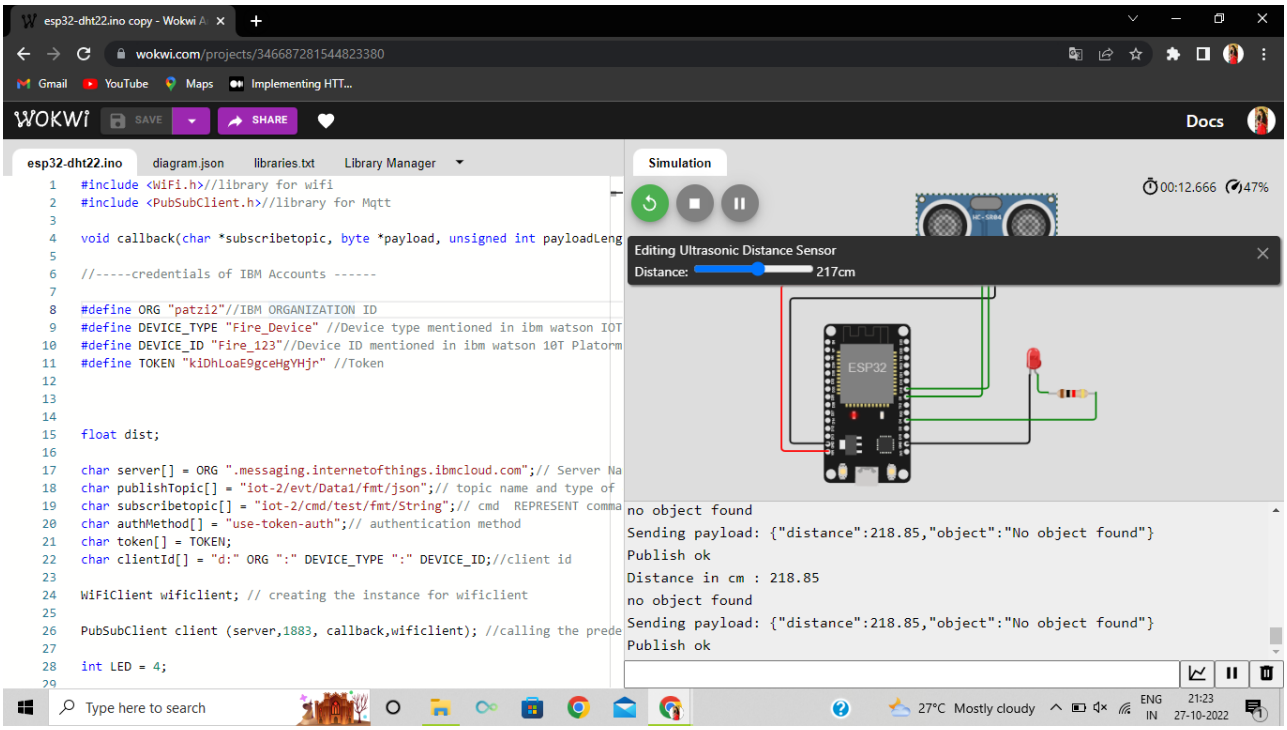
When the distance is less than 100 cms, send an “alert” message to IBM Watson IOT Platform.



IBM CLOUD IMAGE



When the object is far (greater than 100 cms), send “no object found” to the IBM Watson IOT Platform.



IBM CLOUD IMAGE