Assignment -4
Python Programming

Assignment Date	17 November 2022
Student Name	K. Aravinth Raj
Student Roll Number	920819106006
Maximum Marks	2 Marks

Question-1:

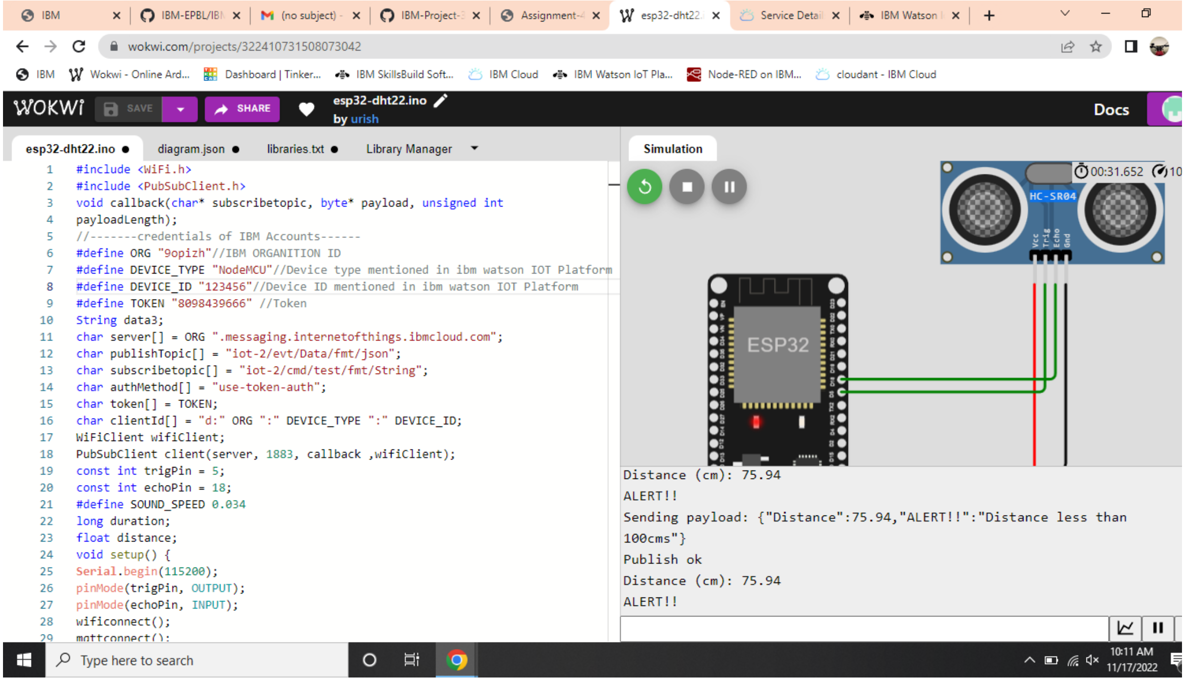
Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud

Solution:

```
#include<WiFi.h>
#include<PubSubClient.h>
voidcallback(char* subscribetopic, byte* payload, unsignedint
payloadLength);


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

#define ORG "90pizh”//IBM ORGANITION ID
#define DEVICE_TYPE "NodeMCU"//Device type mentioned in ibmwatson IOT
Platform
#define DEVICE_ID "123456"//Device ID mentioned in ibmwatson IOT

Platform
#define TOKEN "8098439666"//Token
String data3;
charserver[] = ORG ".messaging.internetofthings.ibmcloud.com";
charpublishTopic[] = "iot-2/evt/Data/fmt/json";
charsubscribetopic[] = "iot-2/cmd/test/fmt/String";
charauthMethod[] = "use-token-auth";
chartoken[] = TOKEN;
charclientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClientwifiClient;
PubSubClientclient(server, 1883, callback ,wifiClient);
constinttrigPin = 5;
constintechoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
voidsetup() {
```

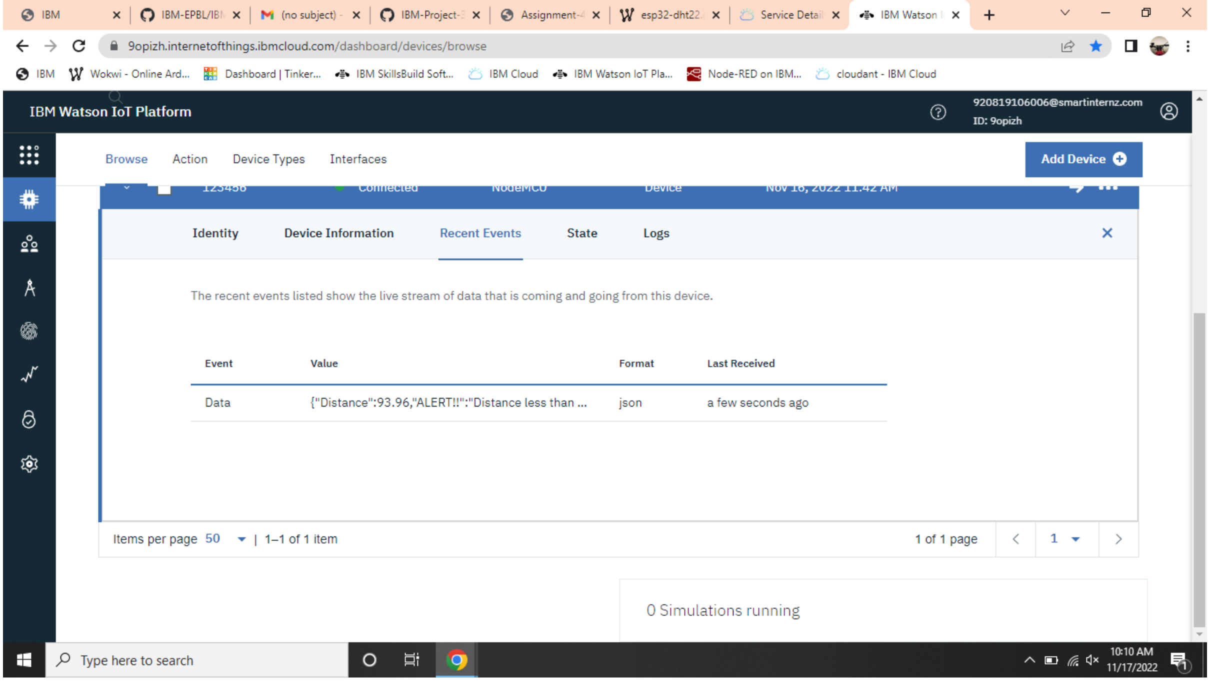
```
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
voidloop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if(!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
voidPublishData(floatdist) {
mqttconnect();
String payload = "{\"Distance\":";
payload += dist;
payload += ",\"ALERT!!\":""\"Distance less than 100cms\"";
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");
}
}
voidmqttconnect() {
if(!client.connected()) {
```

```
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!!!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
voidwificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
voidinitManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
voidcallback(char* subscribetopic, byte* payload, unsignedintpayloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (inti = 0; i<payloadLength; i++) {

//Serial.print((char)payload[i]);

data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}
```



```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "9opizh"//IBM ORGANITION ID
#define DEVICE_TYPE "NodeMCU"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "123456"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "8098439666" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
```

```
Distance (cm): 75.94
ALERT!!
Sending payload: {"Distance":75.94,"ALERT!!":"Distance less than
100cms"}
Publish ok
Distance (cm): 75.94
ALERT!!
```



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":93.96,"ALERT!!":"Distance less than ...	json	a few seconds ago

WOWKI LINK :

wokwi.com/projects/348548028244689491