Assignment -4

ESP32 Programming

Assignment Date	1 November 2022
Student Name	J.JEYANTHI
Student Roll Number	962819106023
Maximum Marks	2 Marks

Question-1:

Write code and connections in wokwi for the ultrasonic sensor.

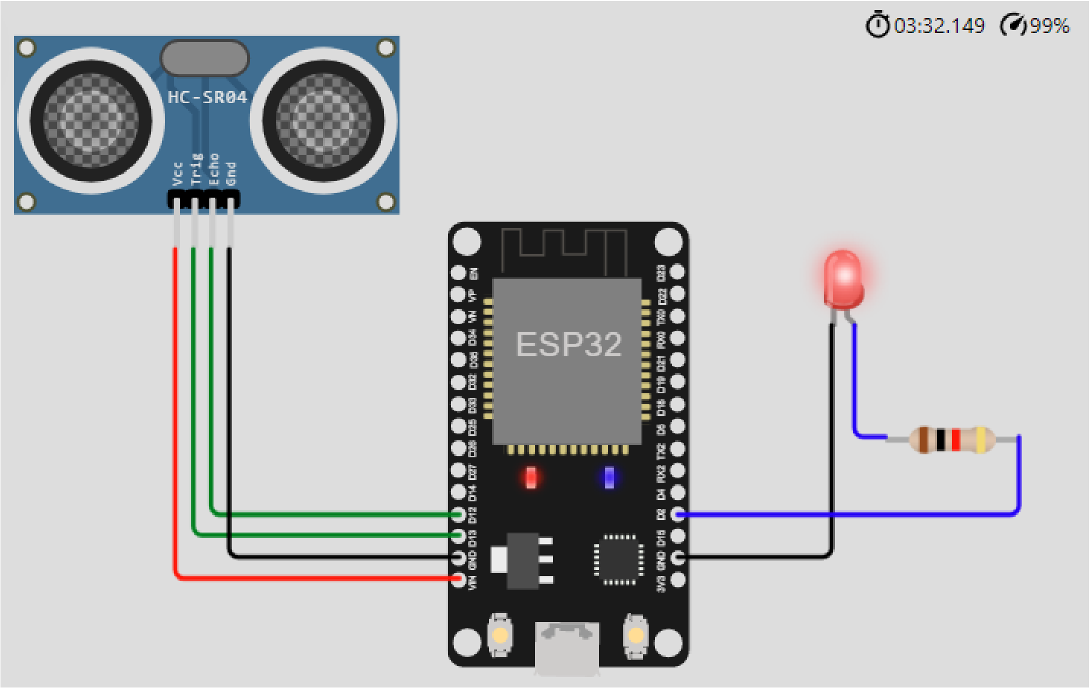
Whenever the distance is less than 100cms send an “alert” to the IBM cloud and display the device recent events.

Upload document with wokwi share link and images of IBM cloud.

My Wokwi share link:

https://wokwi.com/projects/347049779330548308

Circuit:

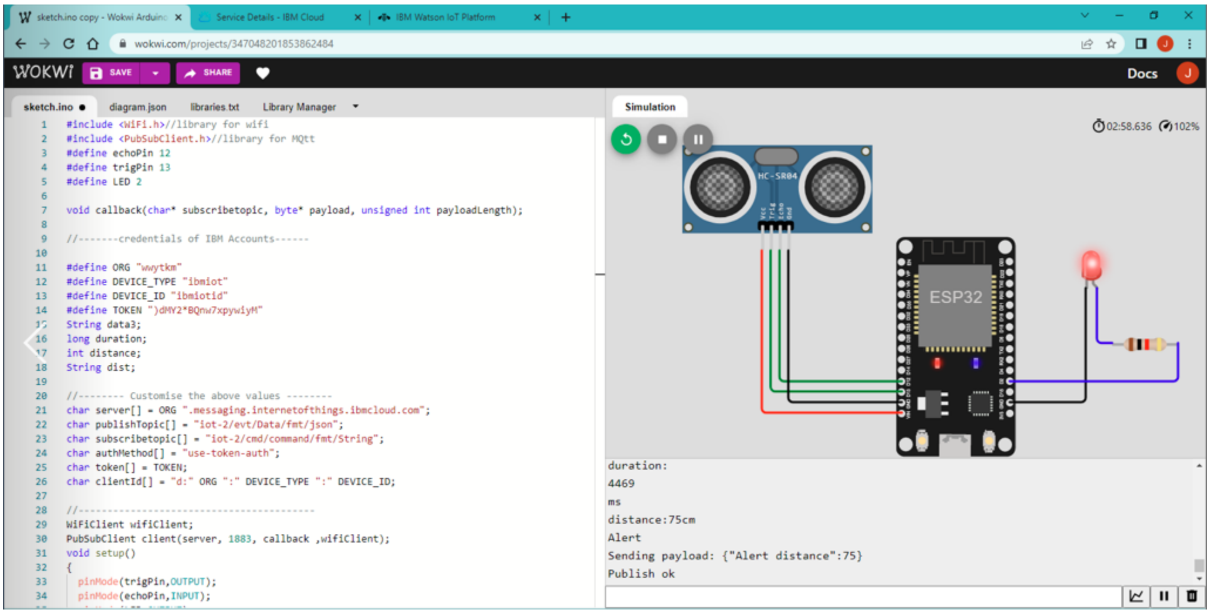


Code:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define echoPin 12
#define trigPin 13
#define LED 2

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

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

#define ORG "wwytkm"
#define DEVICE_TYPE "ibmiot"
#define DEVICE_ID "ibmiotid"
#define TOKEN ")dMY2*BQnw7xpywiyM"
String data3;
long duration;
int distance;
String dist;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/command/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);
void setup()
{
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
  pinMode(LED,OUTPUT);
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()
{
 digitalWrite(trigPin,LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin,LOW);
 duration=pulseIn(echoPin,HIGH);
 Serial.println("duration:");
 Serial.println(duration);
 Serial.println("ms");
 distance=(duration*0.034)/2;
 Serial.println("distance:"+String(distance)+"cm");
 if (distance<100)
 {
   digitalWrite(LED,HIGH);
 }
 else{
   digitalWrite(LED,LOW);
 }
 Serial.println("Alert"+dist);
 if(distance<100)
 {
 PublishData(dist);
 delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
 }
}


/*.....................................retrieving to Cloud...............................*/

void PublishData(String dist) {
  mqttconnect();
  String payload = "{\"Alert distance\":";
  payload += distance;
  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");
  }

}
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()
{
  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());
}
```

```
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);
}
```

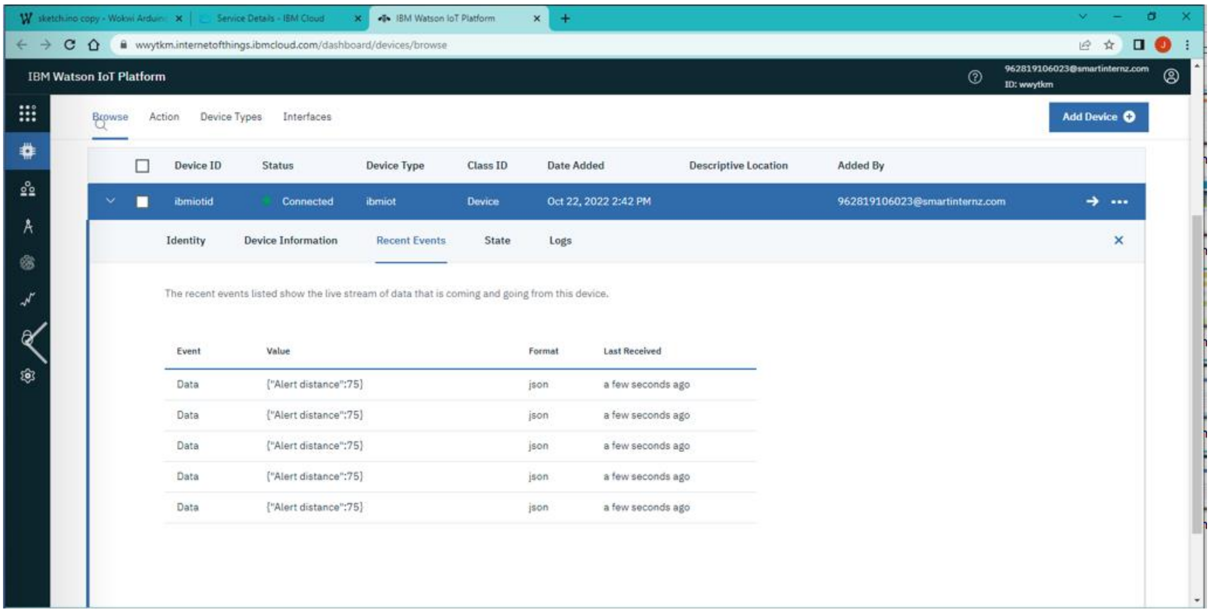
Output:



```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define echoPin 12
#define trigPin 13
#define LED 2

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

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

#define ORG "wwytkm"
#define DEVICE_TYPE "ibmiot"
#define DEVICE_ID "ibmiotid"
#define TOKEN ")dMY2*BQnw7xpywiyM"
String data3;
long duration;
int distance;
String dist;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/command/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);
void setup()
{
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
```

```
duration:
4469
ms
distance:75cm
Alert
Sending payload: {"Alert distance":75}
Publish ok
```

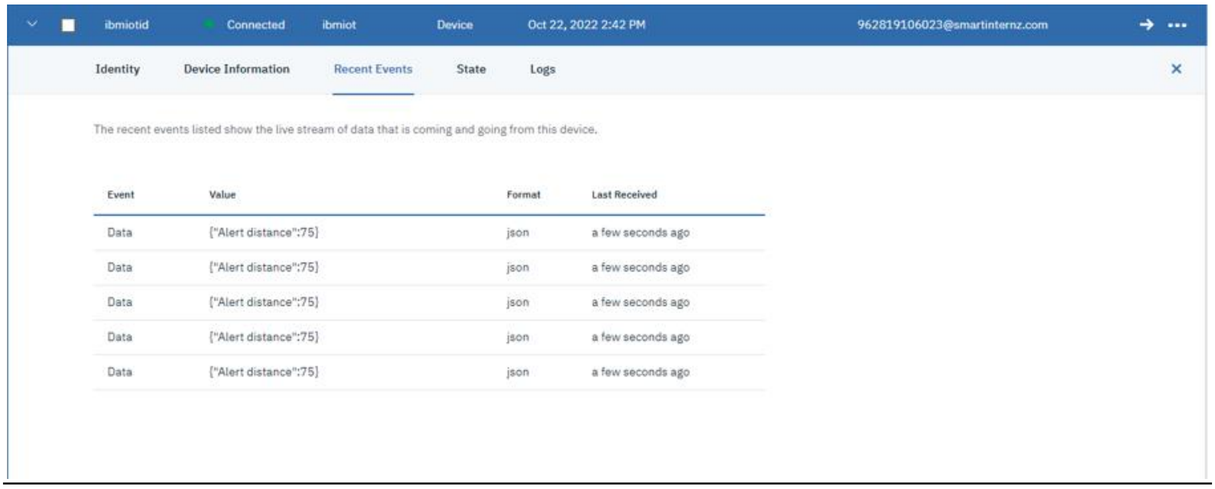
Images of IBM Cloud:


sketch.ino copy - Wokwi Arduin
Service Details - IBM Cloud
IBM Watson IoT Platform
wwytkm.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
962819106023@smartinternz.com
ID: wwytkm
Browse
Action
Device Types
Interfaces
Add Device

□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
□	ibmiotid	● Connected	ibmiot	Device	Oct 22, 2022 2:42 PM		962819106023@smartinternz.com

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

Event	Value	Format	Last Received
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago


□	ibmiotid	● Connected	ibmiot	Device	Oct 22, 2022 2:42 PM	962819106023@smartinternz.com

Identity Device Information Recent Events State Logs

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

Event	Value	Format	Last Received
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago
Data	{"Alert distance":75}	json	a few seconds ago