Write code and connections in wokwi for the ultrasonic sensor.

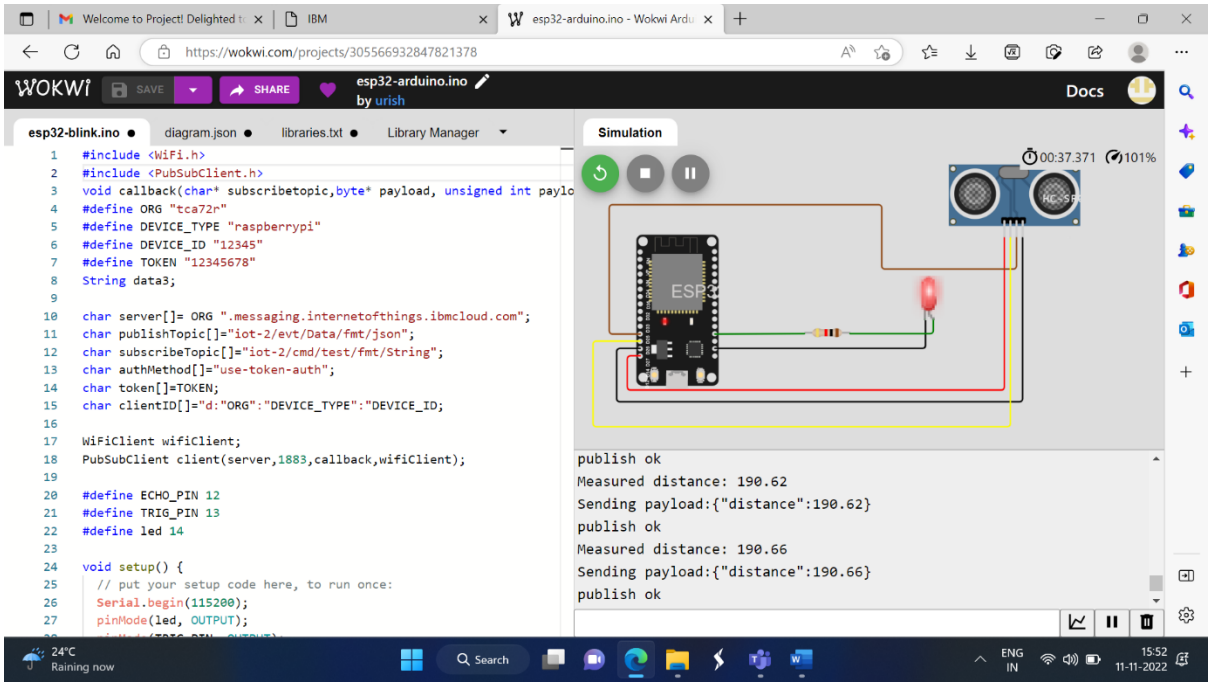
Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

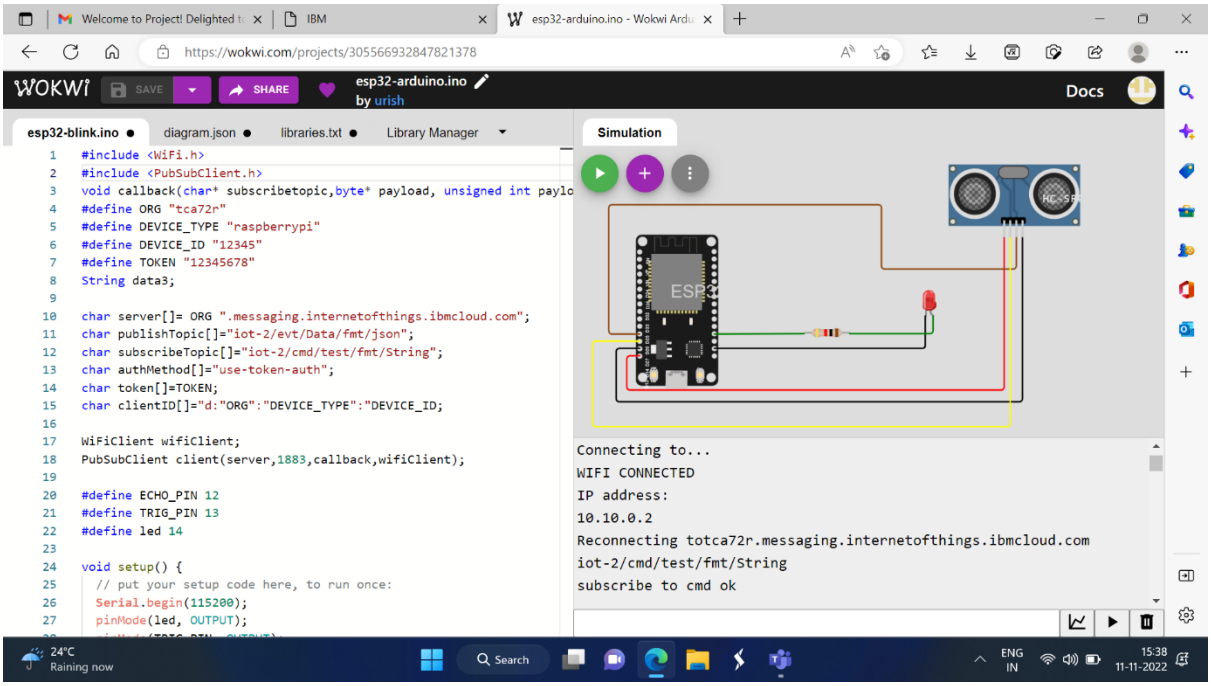
Date	11-11-2022
Team ID	PNT2022TMID28313
Project Name	Industry-specific intelligent fire management system
Maximum Marks	2 Marks

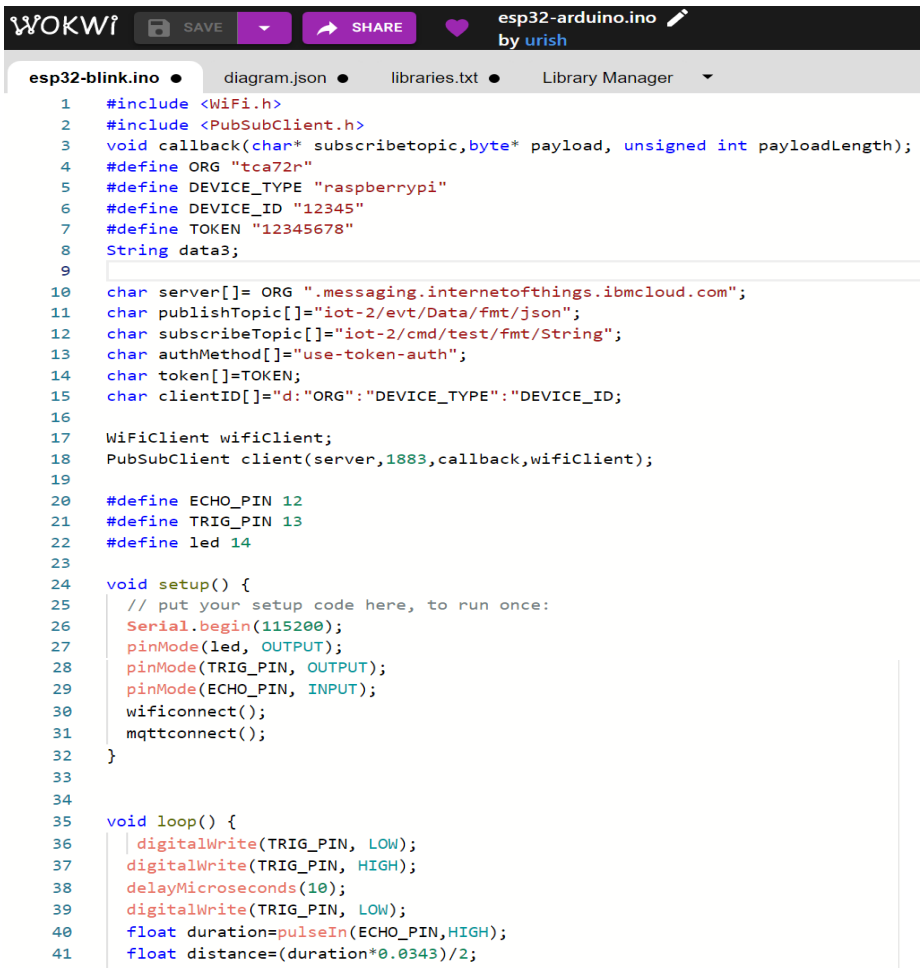
WOKWI WEB URL:

https://wokwi.com/projects/305566932847821378

SIMULATION SCREENSHOT :




Welcome to Project! Delighted t
IBM
esp32-arduino.ino - Wokwi Ardu
https://wokwi.com/projects/305566932847821378
WOKWI
SAVE
SHARE
esp32-arduino.ino
by urish
Docs
esp32-blink.ino
diagram.json
libraries.txt
Library Manager
Simulation
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int paylo
#define ORG "tca72r"
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
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);
#define ECHO_PIN 12
#define TRIG_PIN 13
#define led 14
void setup() {
// put your setup code here, to run once:
Serial.begin(115200);
pinMode(led, OUTPUT);
Connecting to...
WIFI CONNECTED
IP address:
10.10.0.2
Reconnecting totca72r.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd ok
24°C
Raining now
Search
ENG
IN
15:38
11-11-2022


WOKWI
SAVE
SHARE
esp32-arduino.ino
by urish
esp32-blink.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG "tca72r"
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
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);
#define ECHO_PIN 12
#define TRIG_PIN 13
#define led 14
void setup() {
// put your setup code here, to run once:
Serial.begin(115200);
pinMode(led, OUTPUT);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
wificonnect();
mqttconnect();
}
void loop() {
digitalWrite(TRIG_PIN, LOW);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
float duration=pulseIn(ECHO_PIN,HIGH);
float distance=(duration*0.0343)/2;

```
42
43    bool isNearby = distance < 100;
44    digitalWrite(led, isNearby);
45
46    Serial.print("Measured distance: ");
47    Serial.println(distance);
48    if(distance<100){
49      PublishData2(distance);
50
51    }else{
52      PublishData1(distance);
53
54    }
55    //PublishData(distance);
56    delay(1000);
57    if(!client.loop()){
58      mqttconnect();
59    }
60
61    //delay(2000);
62  }
63  void PublishData1(float dist){
64    mqttconnect();
65    String payload= "{\"distance\":";
66    payload += dist;
67    payload+="}";
68
69    Serial.print("Sending payload:");
70    Serial.println(payload);
71
72    if(client.publish(publishTopic,(char*)payload.c_str())){
73      Serial.println("publish ok");
74    } else{
75      Serial.println("publish failed");
76    }
77  }
78  void PublishData2(float dist){
79    mqttconnect();
80    String payload= "{\"ALERT\":";
81    payload += dist;
82    payload+="}";
83
84    Serial.print("Sending payload:");
85    Serial.println(payload);
86
87    if(client.publish(publishTopic,(char*)payload.c_str())){
88      Serial.println("publish ok");
89    } else{
90      Serial.println("publish failed");
91    }
92  }
93  void mqttconnect(){
94    if(!client.connected()){
95      Serial.print("Reconnecting to");
96      Serial.println(server);
97      while(!!!client.connect(clientID, authMethod, token)){
98        Serial.print(".");
99        delay(500);
100     }
101     initManagedDevice();
102     Serial.println();
103   }
104 }
105
106 void wificonnect(){
107   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++){
    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="";
}
```