

Write a 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.

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "13jrby"
#define DEVICE_TYPE "Kowsalya1912"
#define DEVICE_ID "Kowsalya123"
#define TOKEN "kowsi123"
#define speed 0.034
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
int dist;
void setup()
{
  Serial.begin(115200);
```

```

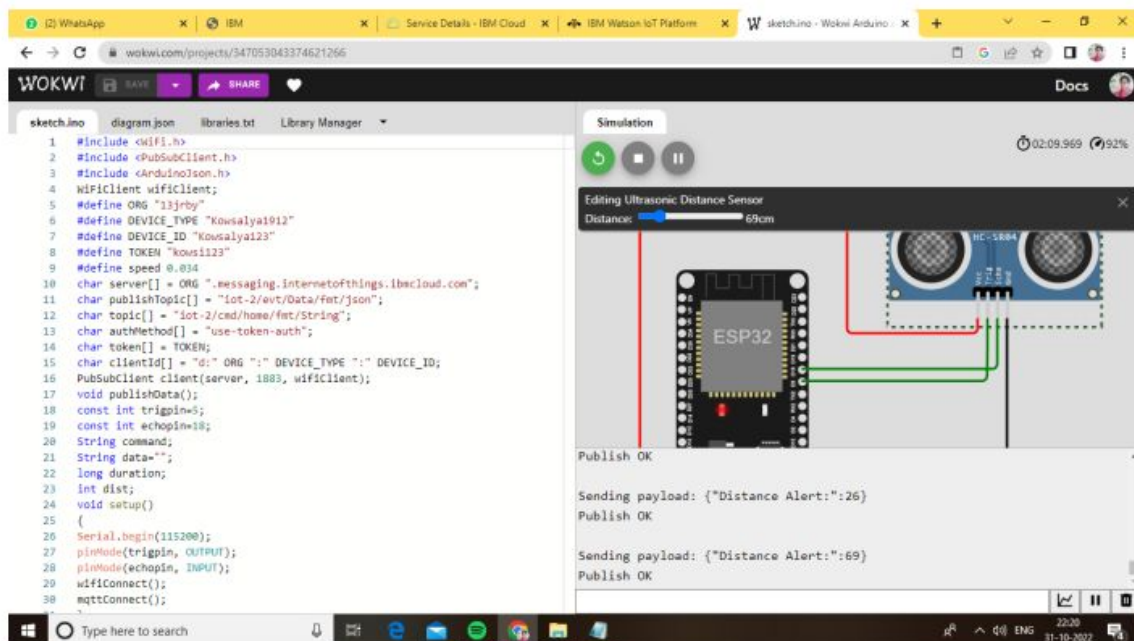
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
}
void loop() {
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}
void wifiConnect() {
Serial.print("Connecting to "); Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect() {
if (!client.connected()) {
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token))
{
Serial.print(".");
delay(1000);
}
initManagedDevice();
Serial.println();
}
}

```

```

}
void initManagedDevice() {
if (client.subscribe(topic)) {
Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin, LOW);
digitalWrite(trigpin, HIGH);
delayMicroseconds(10);
digitalWrite(trigpin, LOW);
duration=pulseIn(echopin, HIGH);
dist=duration*speed/2;
if(dist<100){
DynamicJsonDocument doc(1024);
String payload;
doc["Distance Alert:"]=dist;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*)
payload.c_str())) {
Serial.println("Publish OK");
} else {
Serial.println("Publish FAILED");
}
}
}
}

```



Wokwi Link:

<https://wokwi.com/projects/347053043374621266>

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A user profile is visible in the top right corner with the email '922119104044@smartinternz.com' and the name 'ID: 13jby'. A sidebar on the left contains various icons for navigation. The main content area shows a table of devices with columns: Device ID, Status, Device Type, Class ID, and Date Added. One device, 'Kowsalya123', is highlighted with a status of 'Connected' and a device type of 'Kowsalya1912'. Below this, a 'Recent Events' tab is selected, showing a list of events with columns: Event, Value, Format, and Last Received. The events are JSON strings representing distance alerts.

Device ID	Status	Device Type	Class ID	Date Added
Kowsalya123	Connected	Kowsalya1912	Device	Oct 31, 2022 9:33 PM

Event	Value	Format	Last Received
Data	{"Distance Alert":68}	json	a few seconds ago
Data	{"Distance Alert":69}	json	a few seconds ago
Data	{"Distance Alert":26}	json	a few seconds ago
Data	{"Distance Alert":96}	json	a few seconds ago
Data	{"Distance Alert":58}	json	a minute ago