

**Assignment -4**  
Python Programming

Assignment Date	01 November 2022
Student Name	Mirdhula.M
Student Roll Number	210819106040
Maximum Marks	2 Marks

**Question-1:**

Write 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.

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

**Solution:**

← → ↻

wokwi.com/projects/346566226034557523

🔖 ☆ 🏠 🔴 ⋮

WOKWI

SAVE

SHARE

🔖

Docs

🔴

sketch.ino ▾ diagram.json libraries.txt ▾ Library Manager ▾

Simu

▶

Co  
nn  
ec  
ti  
ng  
to

```
1 #include<WiFi.h> //library for wifi
2 #include<PubSubClient.h> //library for MQTT
3 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
4 //-----credentials of IBM Account-----
5 #define ORG "Izzy6o" // IBM ORGANIZATION ID
6 #define DEVICE_TYPE "iotedeviceproject" //DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
7 #define DEVICE_ID "229714" //DEVICE ID MENTIONED IN IOT WATSON PLATFORM
8 #define TOKEN "24681812" //Token
9 String data3;
10 float dist;
11 //-----customize the above value-----
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //server name
13 char publishstopic[] = "ultrasonic/evt/Data/fmt/json"; //topic name and type of event perform
14 | and format in which data to be send*/
15 char subscribetopic[] = "ultrasonic/cmd/test/fmt/String"; //cmd REPRESENT Command tupe and
16 | COMMAND IS TEST OF FORMAT STRING*/
17 char authMethod[] = "use-token-auth"; //authentication method
18 char token[] = TOKEN;
19 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //CLIENT ID
20 //-----
21 WiFiClient wificlient; // creating an instance for wificlient
22 PubSubClient client(server, 1883, callback, wificlient); //calling the predefined client id
23 | by passing parameter like server id,portand wificredential*/
24 int LED = 4;
25 int trig = 5;
26 int echo = 18;
27 void setup()
28 {
29   Serial.begin(115200);
30   pinMode(trig, OUTPUT);
```

← → ↺ wokwi.com/projects/346566226034557523

WOKWI

SAVE

SHARE

♥

Docs

V

sketch.ino ● diagram.json libraries.txt ● Library Manager

61Serial.println("no object is near");  
62object="Near";  
63}  
64else  
65{  
66digitalWrite(LED,LOW);  
67Serial.println("no object found");  
68object="No";  
69}  
70String payload="{\"distance\":";  
71payload +=dist;  
72payload +=",\" \"object\":\":";  
73payload += object;  
74payload += "\":";  
75  
76Serial.print("Sending payload: ");  
77Serial.println(payload);  
78if(client.publish(publishtopic, (char\*) payload.c\_str())){  
79Serial.println("Publish ok");/\* If its successfully upload data on the cloud then it will print  
80publish ok in serial monitor or else it will print publish failed\*/  
81}  
82else{  
83Serial.println("Publish failed");  
84}  
85  
86void mqttconnect(){  
87if(!client.connected()){  
88Serial.print("Reconnecting client to ");  
89Serial.println(server);  
90while(!client.connect(clientid,authMethod, token)){  
91Serial.print(".");  
92delay(500);  
93

Simu

▶

Co  
nn  
ec  
ti  
ng  
to

← → ↺ wokwi.com/projects/346566226034557523

WOKWI

SAVE

SHARE

♥

Docs

V

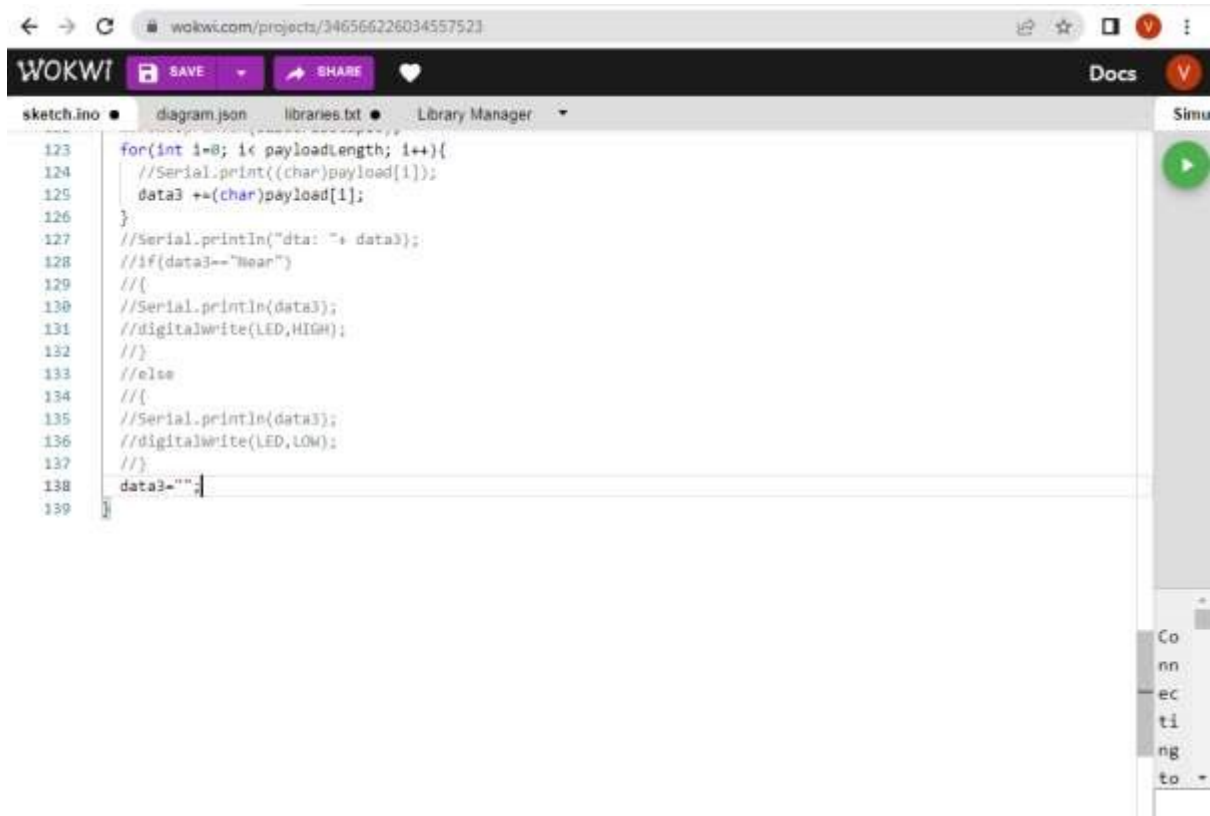
sketch.ino ● diagram.json libraries.txt ● Library Manager

92}  
93initManagedDevice();  
94Serial.println();  
95}  
96}  
97void wificonnect();//function defenition for wificonnect  
98{  
99Serial.println();  
100Serial.print("Connecting to ");  
101WiFi.begin("Wokwi.GUEST", "",6);//PASSING THE WIFI CREDENTIALS TO ESTABLISH CONRECTION  
102while (WiFi.status() !=WL\_CONNECTED){  
103delay(500);  
104Serial.print(".");  
105}  
106Serial.println("");  
107Serial.println("Wifi connected");  
108Serial.println("IP address");  
109Serial.println(WiFi.localIP());  
110}  
111void initManagedDevice(){  
112if(client.subscribe(subscribetopic)){  
113Serial.println((subscribetopic));  
114Serial.println("subscribe to cmd OK");  
115}  
116else{  
117Serial.println("subscribe to cmd failed");  
118}  
119}  
120void callback(char\* subscribetopic,byte\*payload,unsigned int payloadLength)  
121{  
122Serial.print("callback invoked for topic: ");  
123Serial.println(subscribetopic);  
124

Simu

▶

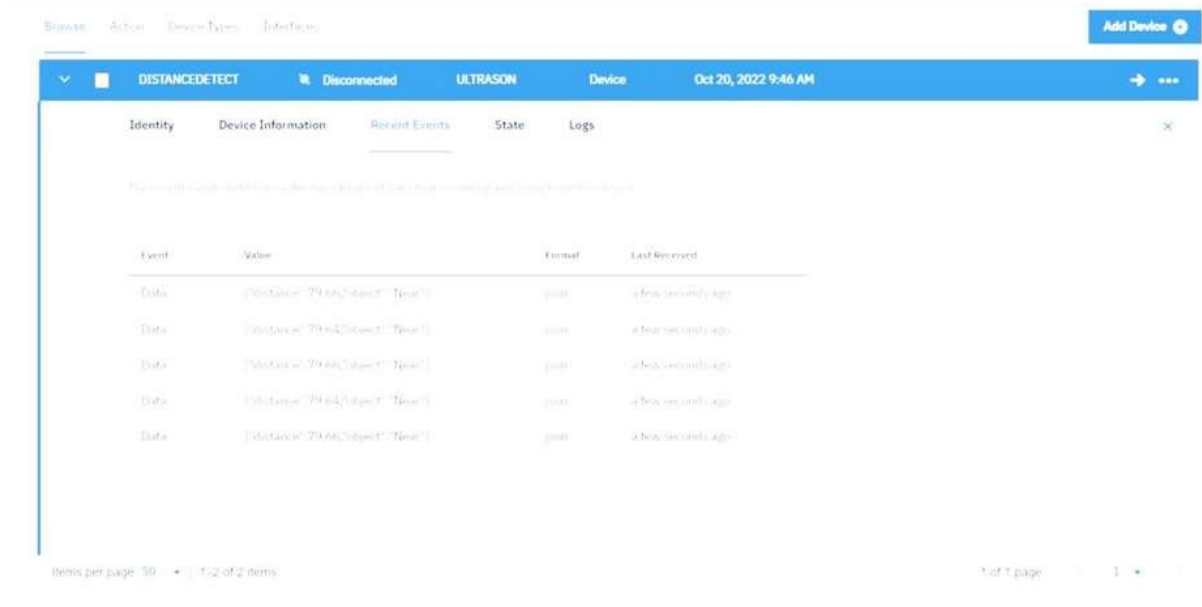
Co  
nn  
ec  
ti  
ng  
to



OUTPUT:  
DATA IS SENT TO IBM CLOUD WHEN NO OBJECT IS DETECTED

DISTANCEDETECT				
Disconnected		ULTRASON	Device	Oct 20, 2022 9:46 AM
Identity	Device Information	Recent Events	State	Logs
<p>Hardware: Arduino Uno, Software: Arduino IDE 1.8.19, Board: Arduino Uno, Compiler: GCC 5.4.0</p>				
Event	Value	Format	Last Received	
Data	[{"distance": 79.66, "object": "Near"}]	json	a few seconds ago	
Data	[{"distance": 79.64, "object": "Near"}]	json	a few seconds ago	
Data	[{"distance": 79.66, "object": "Near"}]	json	a few seconds ago	
Data	[{"distance": 79.64, "object": "Near"}]	json	a few seconds ago	
Data	[{"distance": 79.66, "object": "Near"}]	json	a few seconds ago	

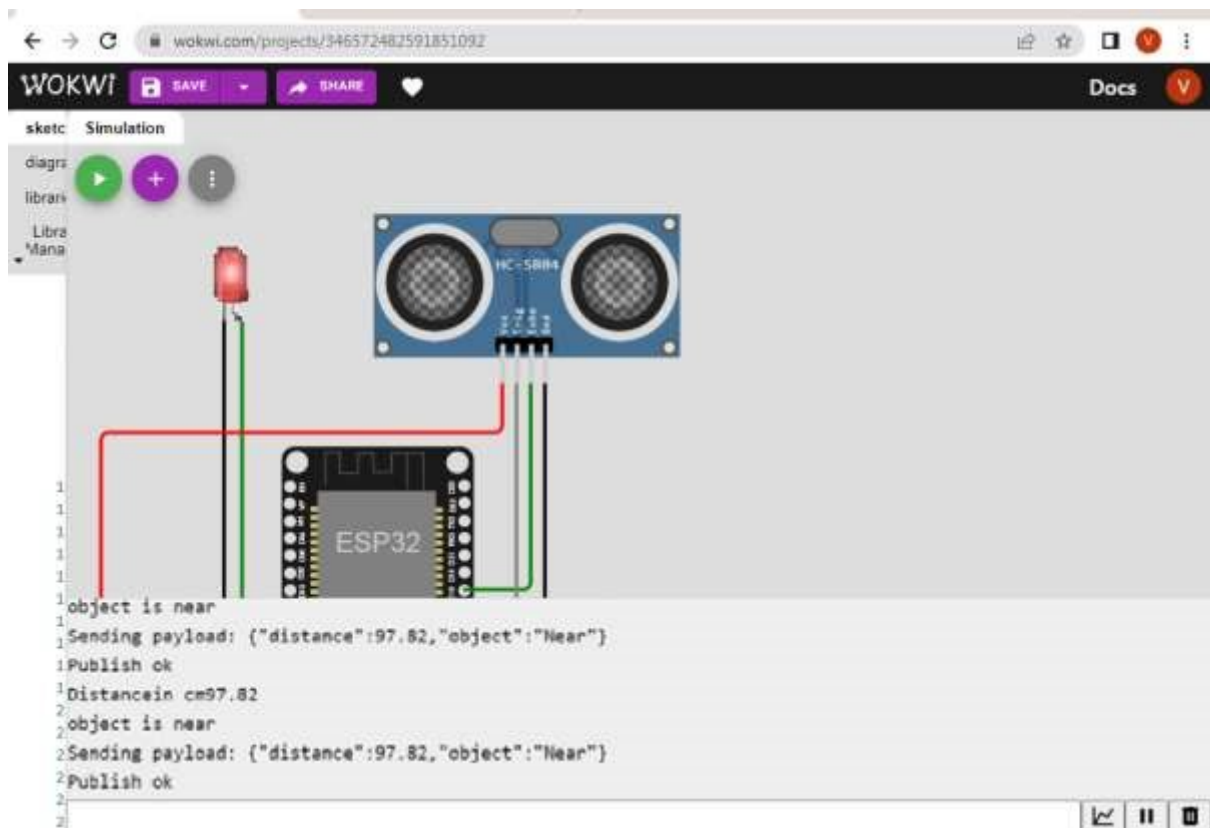
## When no object is detected



The screenshot shows a web interface for a device named 'DISTANCEDETECT'. The device is 'Disconnected' and is of type 'ULTRASON'. The date and time are 'Oct 20, 2022 9:46 AM'. The interface has tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. There are five events listed, all with the value '[Distance: 79.66/object: "Near"]' and the format 'json'. The 'Last Received' column shows 'a few seconds ago' for each event. At the bottom, it says 'Items per page: 50' and '1 of 2 items'.

Event	Value	Format	Last Received
Data	[Distance: 79.66/object: "Near"]	json	a few seconds ago
Data	[Distance: 79.66/object: "Near"]	json	a few seconds ago
Data	[Distance: 79.66/object: "Near"]	json	a few seconds ago
Data	[Distance: 79.66/object: "Near"]	json	a few seconds ago
Data	[Distance: 79.66/object: "Near"]	json	a few seconds ago

## When object is detected in ultrasonic detector



The screenshot shows a Wokwi simulation of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor and a red LED. The sensor is connected to the ESP32 via I2C (VCC to 5V, GND to GND, Trig to SDA, Echo to SCL). The LED is connected to the ESP32 via a digital pin (VCC to 5V, GND to GND, and the LED pin to a digital pin). The simulation is running, and the console shows the following output:

```
1 object is near
1 Sending payload: {"distance":97.82,"object":"Near"}
1 Publish ok
1 Distance in cm 97.82
2 object is near
2 Sending payload: {"distance":97.82,"object":"Near"}
2 Publish ok
```