

Assignment -4
Python Programming

Assignment Date	28 October 2022
Student Name	V.VENKATADRI
Student Roll Number	111719106170
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 SAVE SHARE Docs

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

```

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 "izyy6o" // IBM ORGANIZATION ID
6 #define DEVICE_TYPE "iotdeviceproject" //DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
7 #define DEVICE_ID "229714" //DEVICE ID MENTIONED IN IOT WATSON PLATFORM
8 #define TOKEN "24681012" //Token
9 String data3;
10 float dist;
11 //-----customize the above value-----
12 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //server name
13 char publishtopic[] = "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 type 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,port and 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);

```

Co
nn
ec
ti
ng
to

← → ↺

wokwi.com/projects/346566226034557523

🔖 ☆ 📄 🔴

WOKWI

📄 SAVE ↗️

🔗 SHARE 📄

📄

Docs 🔴

sketch.ino

diagram.json

libraries.txt

Library Manager

Simu

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

Co
nn
ec
ti
ng
to

← → ↺

wokwi.com/projects/346566226034557523

🔖 ☆ 📄 🔴

WOKWI

📄 SAVE ↗️

🔗 SHARE 📄

📄

Docs 🔴

sketch.ino

diagram.json

libraries.txt

Library Manager

Simu

```
92 }
93 initManagedDevice();
94 Serial.println();
95 }
96 }
97 void wificonnect();//function definition for wificonnect
98 {
99   Serial.println();
100   Serial.print("Connecting to ");
101   WiFi.begin("wokwi.GUEST", "",0);//PASSING THE WIFI CREDENTIALS TO ESTABLISH CONNECTION
102   while (WiFi.status() !=WL_CONNECTED){
103     delay(500);
104     Serial.print(".");
105   }
106   Serial.println("");
107   Serial.println("Wifi connected");
108   Serial.println("IP address");
109   Serial.println(WiFi.localIP());
110 }
111 void initManagedDevice(){
112   if(client.subscribe(subscribetopic)){
113     Serial.println((subscribetopic));
114     Serial.println("subscribe to cmd OK");
115   }else{
116     Serial.println("subscribe to cmd failed");
117   }
118 }
119 void callback(char* subscribetopic,byte*payload,unsigned int payloadLength)
120 {
121   Serial.print("callback invoked for topic: ");
122   Serial.println(subscribetopic);
```

Co
nn
ec
ti
ng
to

```
123 for(int i=0; i< payloadLength; i++){
124   //Serial.print((char)payload[i]);
125   data3 +=(char)payload[i];
126 }
127 //Serial.println("data: "+ data3);
128 //if(data3=="Near")
129 //{
130 //Serial.println(data3);
131 //digitalwrite(LED,HIGH);
132 //}
133 //else
134 //{
135 //Serial.println(data3);
136 //digitalwrite(LED,LOW);
137 //}
138 data3="";
139 }
```

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

Device

Action

Device Types

Interfaces

Add Device

▼

■

DISTANCEDetect

Disconnected

ULTRASON

Device

Oct 20, 2022 9:46 AM

→ ...

Identity

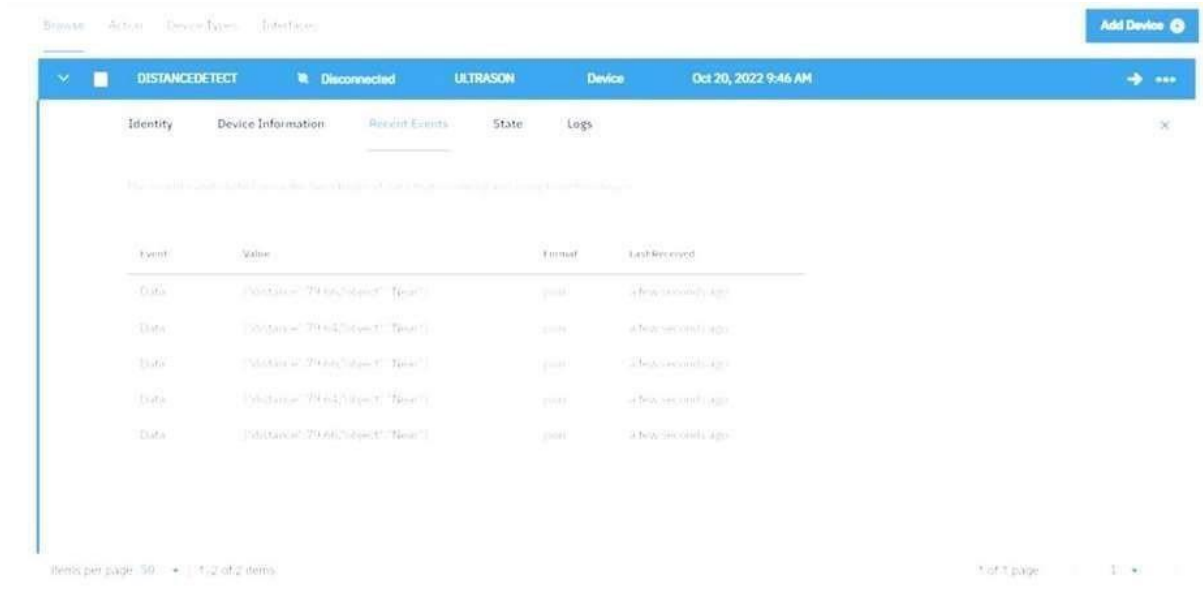
Device Information

Recent Events

State

Logs

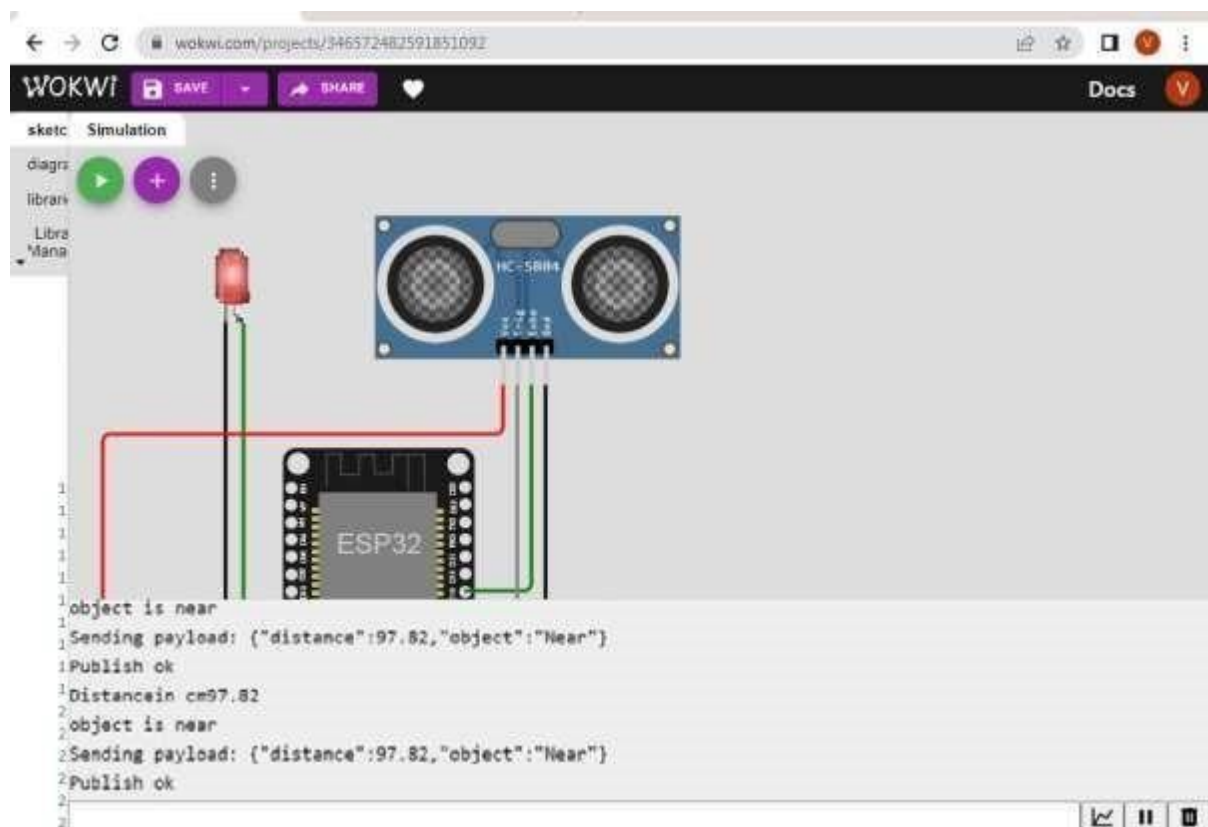
When no object is detected



The screenshot shows the Wokwi web interface for a device named 'DISTANCEDetect'. The device is in a 'Disconnected' state. The 'Events' tab is selected, showing a table of events. The table has columns for 'Event', 'Value', 'Email', and 'Last Received'. The events are all 'Data' events with a value of '{\"distance\":97.82,\"object\":\"Near\"}' and a status of 'good'. The 'Last Received' times are 4 hours, 54 minutes, and 4 seconds ago.

Event	Value	Email	Last Received
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago

When object is detected in ultrasonic detector



The screenshot shows the Wokwi web interface for a project named 'wokwi.com/projects/346572482591851093'. The 'Simulation' tab is selected, showing a circuit diagram of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The sensor is connected to the ESP32 via a breadboard. The sensor's red LED is lit, indicating it is powered on. The 'Events' tab is selected, showing a table of events. The table has columns for 'Event', 'Value', 'Email', and 'Last Received'. The events are all 'Data' events with a value of '{\"distance\":97.82,\"object\":\"Near\"}' and a status of 'good'. The 'Last Received' times are 4 hours, 54 minutes, and 4 seconds ago.

Event	Value	Email	Last Received
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago
Data	{\"distance\":97.82,\"object\":\"Near\"}	good	4 hours, 54 minutes, 4 seconds ago