

ASSIGNMENT 4

ESP 32 AND ULTRASONIC SENSOR DATA SENDS TO IBM CLOUD

PROJECT	IoT ENABLED - REAL TIME WATER QUALITY MONITORING AND CONTROL SYSTEM
NAME	ABDUL FAZIL A
PROJECT ID	PNT2022TMID46026
DATE	22 OCT 2022

QUESTION:

Write a code and connections in wowki for the ultrasonic sensor.

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

Upload documents with wowki share link and images of IBM cloud

PROGRAM:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "5vyvsk"//IBM ORGANITION ID
#define DEVICE_TYPE "REAL_TIME_WATER_QUALITY"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "PNT2022TMID47600"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "8vsl8edAhw2kYeYulZ" //Token
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);
const int trigPin = 2;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
```

```

pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(float dist) {
mqttconnect();
String payload = "{\"Distance\": ";
payload += dist;
payload += ", \"ALERT!!\": \"\" \"Distance less than 100cms\"";
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);
  data3="";
}

```

OUTPUT:

WOKWI

SAVE SHARE

ultrasonic with arduino.ino copy

Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

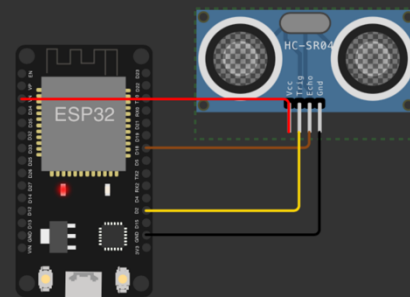
```
60 Serial.print("Sending payload: ");
61 Serial.println(payload);
62
63 if (client.publish(publishTopic, (char*) payload.c_str())) {
64   Serial.println("Publish ok");
65 } else {
66   Serial.println("Publish failed");
67 }
68
69 void mqttconnect() {
70   if (!client.connected()) {
71     Serial.print("Reconnecting client to ");
72     Serial.println(server);
73     while (!client.connect(clientId, authMethod, token)) {
74       Serial.print(".");
75       delay(500);
76     }
77     initManagedDevice();
78     Serial.println();
79   }
80 }
81 void wificonnect()
82 {
83   Serial.println();
84   Serial.print("Connecting to ");
85   WiFi.begin("Wokwi-GUEST", "", 6);
86   while (WiFi.status() != WL_CONNECTED) {
87     delay(500);
88     Serial.print(".");
89   }
90   Serial.println("");
91   Serial.println("WiFi connected");
92   Serial.println("IP address: ");
93   Serial.println(WiFi.localIP());
94 }
95 void initManagedDevice() {
96   if (client.subscribe(subscribetopic)) {
97     Serial.println((subscribetopic));
98     Serial.println("subscribe to cmd OK");
99   } else {
```

Simulation

00:47.639 95%

Editing Ultrasonic Distance Sensor

Distance: 51cm



less than 100cms"}
Publish ok
Distance (cm): 51.99
ALERT!!
Sending payload: {"Distance":51.99,"ALERT!!":"Distance
less than 100cms"}
Publish ok

WOKWI

SAVE SHARE

ultrasonic with arduino.ino copy

Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

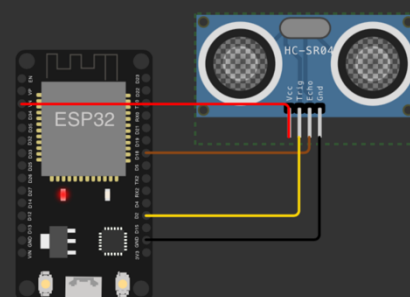
```
60 Serial.print("Sending payload: ");
61 Serial.println(payload);
62
63 if (client.publish(publishTopic, (char*) payload.c_str())) {
64   Serial.println("Publish ok");
65 } else {
66   Serial.println("Publish failed");
67 }
68
69 void mqttconnect() {
70   if (!client.connected()) {
71     Serial.print("Reconnecting client to ");
72     Serial.println(server);
73     while (!client.connect(clientId, authMethod, token)) {
74       Serial.print(".");
75       delay(500);
76     }
77     initManagedDevice();
78     Serial.println();
79   }
80 }
81 void wificonnect()
82 {
83   Serial.println();
84   Serial.print("Connecting to ");
85   WiFi.begin("Wokwi-GUEST", "", 6);
86   while (WiFi.status() != WL_CONNECTED) {
87     delay(500);
88     Serial.print(".");
89   }
90   Serial.println("");
91   Serial.println("WiFi connected");
92   Serial.println("IP address: ");
93   Serial.println(WiFi.localIP());
94 }
95 void initManagedDevice() {
96   if (client.subscribe(subscribetopic)) {
97     Serial.println((subscribetopic));
98     Serial.println("subscribe to cmd OK");
99   } else {
```

Simulation

00:48.805 98%

Editing Ultrasonic Distance Sensor

Distance: 51cm



less than 100cms"}
Publish ok
Distance (cm): 51.99
ALERT!!
Sending payload: {"Distance":51.99,"ALERT!!"
less than 100cms"}
Publish ok

WOKWI SAVE SHARE ultrasonic with arduino.ino copy Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

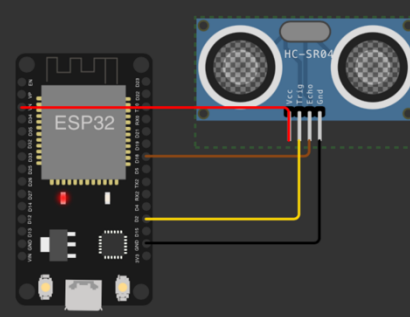
```
60 Serial.print("Sending payload: ");
61 Serial.println(payload);
62
63 if (client.publish(publishTopic, (char*) payload.c_str())) {
64   Serial.println("Publish ok");
65 } else {
66   Serial.println("Publish failed");
67 }
68 }
69
70 void mqttconnect() {
71   if (!client.connected()) {
72     Serial.print("Reconnecting client to ");
73     Serial.println(server);
74     while (!client.connect(clientId, authMethod, token)) {
75       Serial.print(".");
76       delay(500);
77     }
78     initManagedDevice();
79     Serial.println();
80   }
81   void wificonnect()
82   {
83     Serial.println();
84     Serial.print("Connecting to ");
85     WiFi.begin("Wokwi-GUEST", "", 6);
86     while (WiFi.status() != WL_CONNECTED) {
87       delay(500);
88       Serial.print(".");
89     }
90     Serial.println("");
91     Serial.println("WiFi connected");
92     Serial.println("IP address: ");
93     Serial.println(WiFi.localIP());
94   }
95   void initManagedDevice() {
96     if (client.subscribe(subscribetopic)) {
97       Serial.println((subscribetopic));
98       Serial.println("subscribe to cmd OK");
99     } else {
```

Simulation

01:05.719 95%

Editing Ultrasonic Distance Sensor

Distance: 297cm



Distance (cm): 296.96
Distance (cm): 296.92
Distance (cm): 296.92
Distance (cm): 296.94
Distance (cm): 296.92
Distance (cm): 296.92
Distance (cm): 296.94

WOKWI SAVE SHARE ultrasonic with arduino.ino copy Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

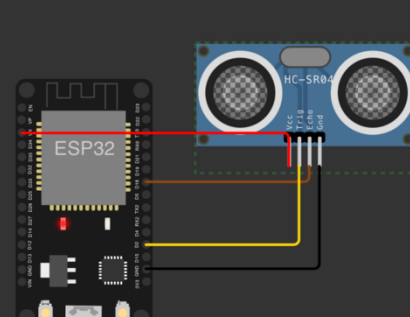
```
60 Serial.print("Sending payload: ");
61 Serial.println(payload);
62
63 if (client.publish(publishTopic, (char*) payload.c_str())) {
64   Serial.println("Publish ok");
65 } else {
66   Serial.println("Publish failed");
67 }
68 }
69
70 void mqttconnect() {
71   if (!client.connected()) {
72     Serial.print("Reconnecting client to ");
73     Serial.println(server);
74     while (!client.connect(clientId, authMethod, token)) {
75       Serial.print(".");
76       delay(500);
77     }
78     initManagedDevice();
79     Serial.println();
80   }
81   void wificonnect()
82   {
83     Serial.println();
84     Serial.print("Connecting to ");
85     WiFi.begin("Wokwi-GUEST", "", 6);
86     while (WiFi.status() != WL_CONNECTED) {
87       delay(500);
88       Serial.print(".");
89     }
90     Serial.println("");
91     Serial.println("WiFi connected");
92     Serial.println("IP address: ");
93     Serial.println(WiFi.localIP());
94   }
95   void initManagedDevice() {
96     if (client.subscribe(subscribetopic)) {
97       Serial.println((subscribetopic));
98       Serial.println("subscribe to cmd OK");
99     } else {
```

Simulation

01:06.798 94%

Editing Ultrasonic Distance Sensor

Distance: 297cm



Distance (cm): 296.92
Distance (cm): 296.92
Distance (cm): 296.94
Distance (cm): 296.92
Distance (cm): 296.92
Distance (cm): 296.94
Distance (cm): 296.92

IBM Watson IoT platform:

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The left sidebar contains various icons for navigation. The main content area displays the 'Recent Events' tab for a specific device. The device information at the top indicates it is 'Disconnected' and has a 'REAL_TIME_WATER_QUALITY' device type. Below this, a table lists recent events.

Event	Value	Format	Last Received
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago

This screenshot shows the same IBM Watson IoT Platform interface, but with the 'Device ID' tab selected. The device information at the top remains the same. The 'Recent Events' tab is still visible below the device information, showing the same table of events as the previous screenshot.

Event	Value	Format	Last Received
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":51.99,"ALERT!!":"Distance less than ...	json	a few seconds ago

Wokwi link:

<https://wokwi.com/projects/348689478640992850>