

Assignment -4

Assignment Date	28 October 2022
Student Name	VE Niranjana
Student Roll Number	2019504556
Maximum Marks	2 Marks

Question :

Write Code and Connections in Wokwi for Ultrasonic Sensor. Whenever Distance is less than 100 cm send "Alert" to IBM Cloud and Display in Device Recent Events.

Source Code :

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG "7r6cyv"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Niranjana_2000"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "9840255557" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/distance/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 2
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();
}
float readDistanceCM() {
digitalWrite(TRIG_PIN, LOW);// Clear the trigger
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);// Sets the trigger pin to HIGH state for 10
microseconds
delayMicroseconds(10);
```

```

digitalWrite(TRIG_PIN, LOW);
int duration=pulseIn(ECHO_PIN, HIGH);
//Serial.println(duration);
//duration = pulseIn(ECHO_PIN, HIGH);
return duration*0.017;
//Serial.println(duration);
}
void loop() {
float distance = readDistanceCM();
//Serial.println(distance);
bool isNearby = distance < 100;
digitalWrite(led, isNearby);
Serial.print("Measured distance: ");
Serial.println(distance);
if(distance<100){
PublishData2(distance);
}else{
PublishData1(distance);
}
//PublishData(distance);
delay(1000);
if(!client.loop()){
mqttconnect();
}
//delay(2000);
}
void PublishData1(float dist){
mqttconnect();
String payload= "{\"distance\":";
payload += dist;
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 PublishData2(float dist){
mqttconnect();
String payload= "{\"ALERT\":";
payload += dist;
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 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++){
data3 += (char)payload[i];
}
Serial.println("data:" + data3);
if(data3=="lighton"){
Serial.println(data3);
digitalWrite(led,HIGH);
}else{
Serial.println(data3);
}
}

```

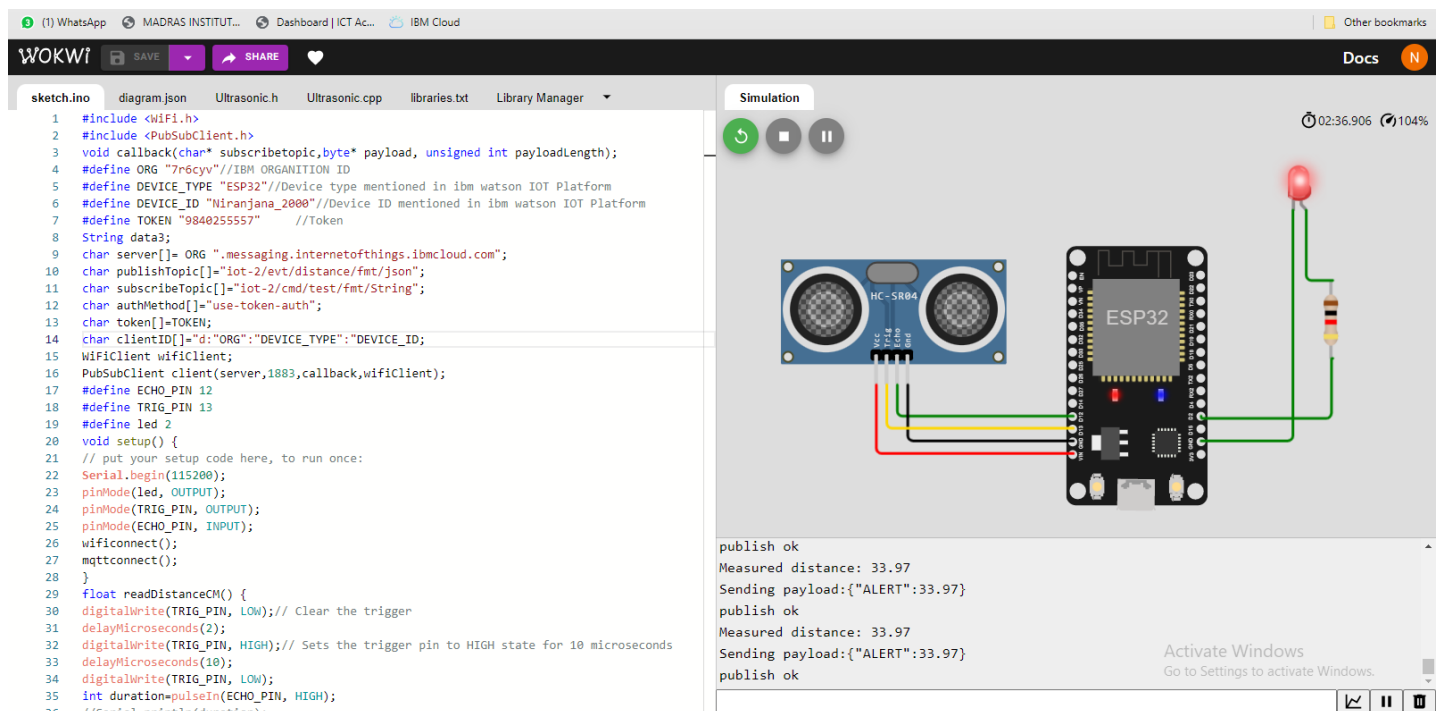
```
digitalWrite(led, LOW);
}
data3="";
}
```

Reference :

<https://wokwi.com/projects/346702474649797204>

Output:

1) Distance less than 100cm – LED Bulb Glows and ‘Alert’ Message is Displayed along with Distance



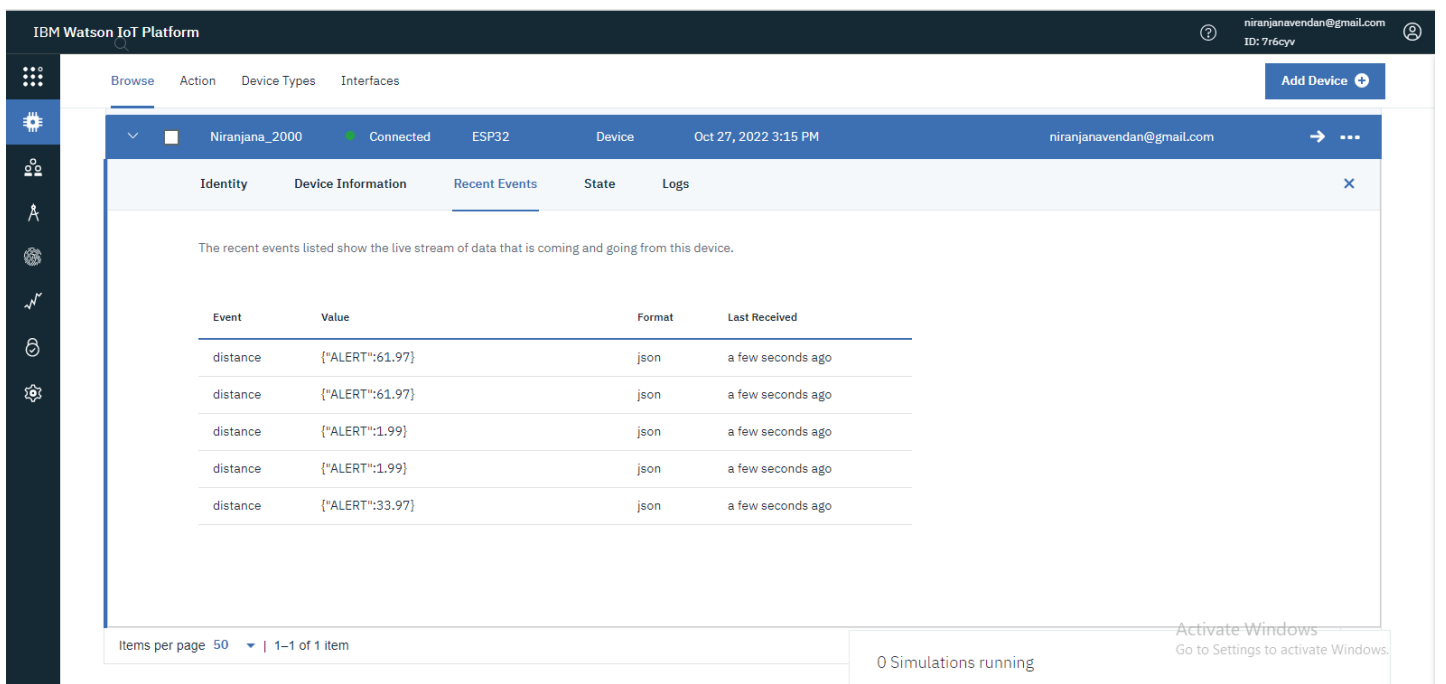
The screenshot shows the Wokwi simulation environment. On the left, the sketch.ino file is open, displaying the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "7r6cyv" // IBM ORGANIZATION ID
5 #define DEVICE_TYPE "ESP32" // Device type mentioned in IBM Watson IoT Platform
6 #define DEVICE_ID "Niranjana_2000" // Device ID mentioned in IBM Watson IoT Platform
7 #define TOKEN "984025557" // Token
8 String data3;
9 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
10 char publishTopic[] = "iot-2/evt/distance/fmt/json";
11 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
15 WiFiClient wifiClient;
16 PubSubClient client(server, 1883, callback, wifiClient);
17 #define ECHO_PIN 12
18 #define TRIG_PIN 13
19 #define led 2
20 void setup() {
21 // put your setup code here, to run once:
22 Serial.begin(115200);
23 pinMode(led, OUTPUT);
24 pinMode(TRIG_PIN, OUTPUT);
25 pinMode(ECHO_PIN, INPUT);
26 wifiConnect();
27 mqttConnect();
28 }
29 float readDistanceCM() {
30 digitalWrite(TRIG_PIN, LOW); // Clear the trigger
31 delayMicroseconds(2);
32 digitalWrite(TRIG_PIN, HIGH); // Sets the trigger pin to HIGH state for 10 microseconds
33 delayMicroseconds(10);
34 digitalWrite(TRIG_PIN, LOW);
35 int duration = pulseIn(ECHO_PIN, HIGH);
36 // Serial.println(duration);
}
```

On the right, the simulation is running. The console output shows the following messages:

```
publish ok
Measured distance: 33.97
Sending payload: {"ALERT":33.97}
publish ok
Measured distance: 33.97
Sending payload: {"ALERT":33.97}
publish ok
```

The simulation also shows a red LED bulb glowing, indicating that the distance is less than 100cm.



The screenshot shows the IBM Watson IoT Platform dashboard. The device 'Niranjana_2000' is listed as 'Connected' with an ESP32 device type. The 'Recent Events' tab is selected, showing a table of events:

Event	Value	Format	Last Received
distance	{"ALERT":61.97}	json	a few seconds ago
distance	{"ALERT":61.97}	json	a few seconds ago
distance	{"ALERT":1.99}	json	a few seconds ago
distance	{"ALERT":1.99}	json	a few seconds ago
distance	{"ALERT":33.97}	json	a few seconds ago

The dashboard also shows the user 'niranjanavendan@gmail.com' and the device ID '7r6cyv'.

2) Distance more than 100cm – LED Bulb OFF and Distance is Displayed

WOKWI

SAVE SHARE

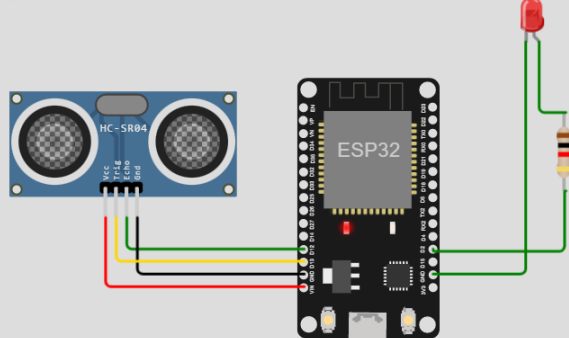
Docs

sketch.ino diagram.json Ultrasonic.h Ultrasonic.cpp libraries.txt Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "7r6cyv" // IBM ORGANITION ID
5 #define DEVICE_TYPE "ESP32" // Device type mentioned in ibm watson IOT Platform
6 #define DEVICE_ID "Niranjana_2000" // Device ID mentioned in ibm watson IOT Platform
7 #define TOKEN "9840255557" // Token
8 String data;
9 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
10 char publishTopic[] = "iot-2/evt/distance/fmt/json";
11 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15 WiFiClient wifiClient;
16 PubSubClient client(server, 1883, callback, wifiClient);
17 #define ECHO_PIN 12
18 #define TRIG_PIN 13
19 #define led 2
20 void setup() {
21 // put your setup code here, to run once:
22 Serial.begin(115200);
23 pinMode(led, OUTPUT);
24 pinMode(TRIG_PIN, OUTPUT);
25 pinMode(ECHO_PIN, INPUT);
26 wifiConnect();
27 mqttConnect();
28 }
29 float readDistanceCM() {
30 digitalWrite(TRIG_PIN, LOW); // Clear the trigger
31 delayMicroseconds(2);
32 digitalWrite(TRIG_PIN, HIGH); // Sets the trigger pin to HIGH state for 10 microseconds
33 delayMicroseconds(10);
34 digitalWrite(TRIG_PIN, LOW);
35 int duration = pulseIn(ECHO_PIN, HIGH);
36 // Serial.println(duration);
}
```

Simulation

01:24.677 81%



publish ok
Measured distance: 361.96
Sending payload: {"distance": 361.96}
publish ok
Measured distance: 361.96
Sending payload: {"distance": 361.96}
publish ok

Activate Windows
Go to Settings to activate Windows.

IBM Watson IoT Platform

niranjanavendan@gmail.com ID: 7r6cyv

Browse Action Device Types Interfaces

Add Device

Niranjana_2000 Connected ESP32 Device Oct 27, 2022 3:15 PM niranjanavendan@gmail.com

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
distance	{"distance": 258.93}	json	a few seconds ago
distance	{"distance": 311.95}	json	a few seconds ago
distance	{"distance": 311.97}	json	a few seconds ago
distance	{"distance": 361.96}	json	a few seconds ago
distance	{"distance": 361.96}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

0 Simulations running

Activate Windows
Go to Settings to activate Windows.