Assignment – 4

Working on Wokwi

Assignment Date	1 November 2022
Student Name	Gowtham Easwar K
Student Roll Number	718019L115
Maximum Marks	2 Marks

Question :

Write a code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an "Alert" to IBM cloud and display in the device recent events.

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "cr4s7d"//IBM ORGANITION ID
#define DEVICE_TYPE "nodeMcu"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "123456"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "123456789" //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 = 5;
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="";
}
```

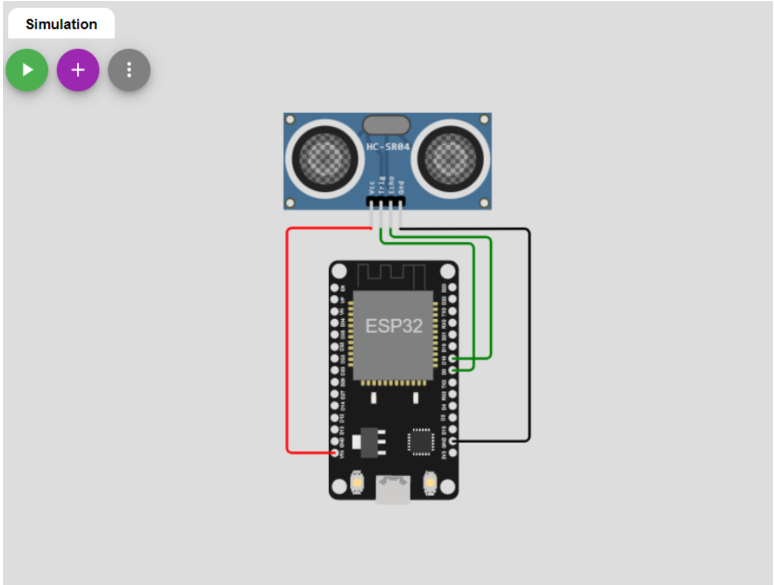
Diagram.json:

```
{
  "version": 1,
  "author": "19L104 - ARAVIND T",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -118.37, "left": -37.17, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v-0.76", "h-68.44", "v184.67" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v-0.1", "h105.22", "v171.33" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "green", [ "v11.9", "h75.67", "v104.67" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v6.57", "h81.78", "v96" ] ]
  ]
}
```

Wokwi Simulation link:

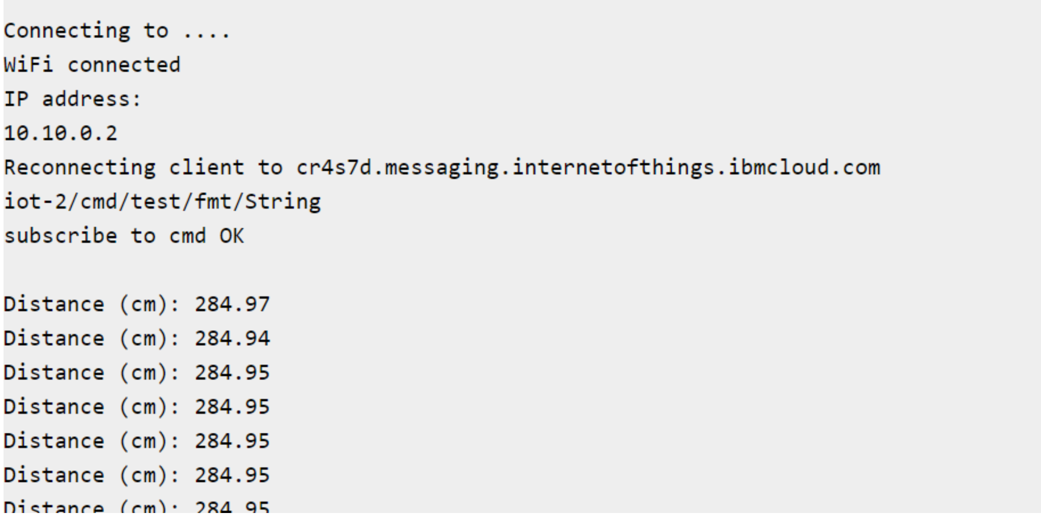
https://wokwi.com/projects/348488317522674258

Circuit Diagram:



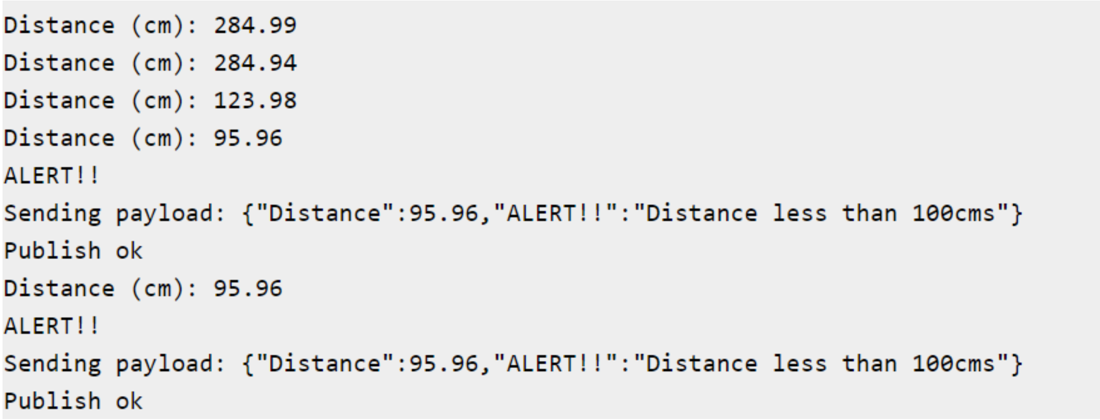
Output:

Wokwi Output



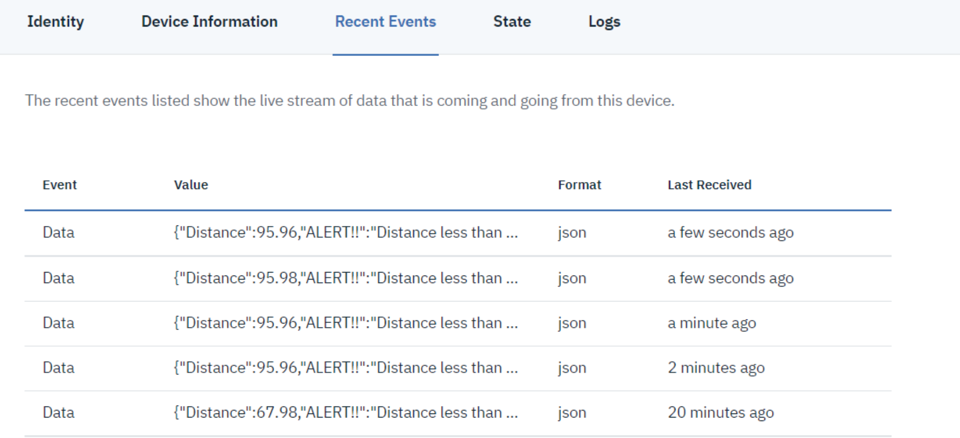
```
Connecting to ....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to cr4s7d.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance (cm): 284.97
Distance (cm): 284.94
Distance (cm): 284.95
Distance (cm): 284.95
Distance (cm): 284.95
Distance (cm): 284.95
Distance (cm): 284.95
```



```
Distance (cm): 284.99
Distance (cm): 284.94
Distance (cm): 123.98
Distance (cm): 95.96
ALERT!!
Sending payload: {"Distance":95.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 95.96
ALERT!!
Sending payload: {"Distance":95.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
```

IBM Cloud Output:



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

Event	Value	Format	Last Received
Data	{"Distance":95.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":95.98,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":95.96,"ALERT!!":"Distance less than ...	json	a minute ago
Data	{"Distance":95.96,"ALERT!!":"Distance less than ...	json	2 minutes ago
Data	{"Distance":67.98,"ALERT!!":"Distance less than ...	json	20 minutes ago