

ASSIGNMENT 4

CODE AND CONNECTIONS FOR ULTRASONIC SENSOR IN WOKWI

QUESTION:

Write code and connections in wokwi for the ultrasonic sensor. whenever the distance is less than 100 cms send an “alert” to the IBM cloud and display in the device recent events. upload document with wokwi share link and images of IBM cloud.

Code:

```
#include <WiFi.h>
#include <PubSubClient.h> void callback(char* subscribetopic,
byte* payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "4tom63"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//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": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 8.67, "left":
115.33, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -36.7, "left":
67.17, "attrs": {} }
  ],
  "connections": [

```

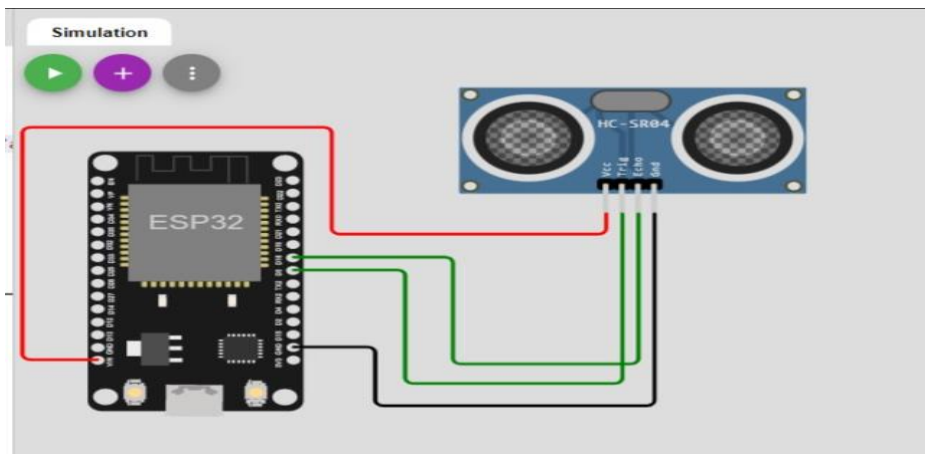
```

[ "esp:TX0", "$serialMonitor:RX", "", [] ],
[ "esp:RX0", "$serialMonitor:TX", "", [] ],
[ "esp:GND.1", "ultrasonic1:GND", "black", [ "h39.87", "v44.04", "h170" ]
],
[ "esp:D18", "ultrasonic1:ECHO", "green", [ "h77.87", "v80.01", "h110" ] ],
[ "esp:D5", "ultrasonic1:TRIG", "green", [ "h54.54", "v85.07", "h130.67" ]
],
[
  [
    "ultrasonic1:VCC",
    "esp:VIN",
    "red",
    [ "v15.24", "h-134.45", "v-80", "h-151.33", "v173.33" ]
  ]
]
]
}

```

Wokwi link: <https://wokwi.com/projects/346736327045153363>

Circuit



Output

```
Connecting to ....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 4tom63.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance (cm): 87.02
ALERT!!
Sending payload: {"Distance":87.02,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 86.96
ALERT!!
Sending payload: {"Distance":86.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 135.97
Distance (cm): 135.93
```

Activate Windows
Go to Settings to activate Windows.

IBM Cloud Output

IBM Watson IoT Platform

kit23.19bec017@gmail.com
ID: 4tom63

Browse Action Device Types Interfaces

Add Device

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
12345	Disconnected	ESP32	Device	Oct 29, 2022 3:58 PM

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
event_1	{"Distance":92,"Alert":"Distance less than 100"}	json	a few seconds ago
event_1	{"Distance":85,"Alert":"Distance less than 100"}	json	a minute ago
event_1	{"Distance":18,"Alert":"Distance less than 100"}	json	2 minutes ago
event_1	{"Distance":74,"Alert":"Distance less than 100"}	json	3 minutes ago
event_1	{"Distance":57,"Alert":"Distance less than 100"}	json	4 minutes ago

1 Simulation running

Activate Windows
Go to Settings to activate Windows.