Assignment -4

Question-1:

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.

Solution:
PROGRAM

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "rv07c6"//IBM ORGANITION ID
#define DEVICE_TYPE "distance_hcsr04"//Device type mentioned in ibm watson IOT
#define DEVICE_ID "6789"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "w_mwV+5NZn*W7Xt)qA" //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 = random(200);
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()
```

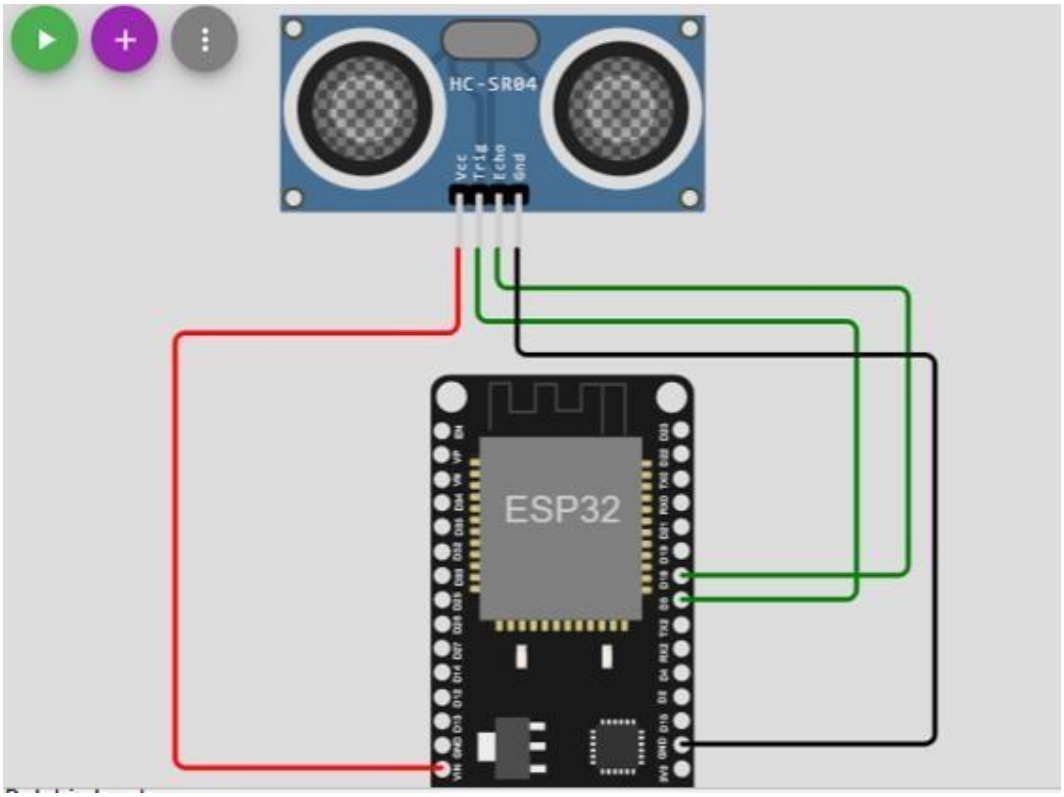
```
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
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="";
}
```

.json CODE:

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

```
{
  "version": 1,
  "author": "Nandhini Mohan",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 69.33, "left": 6.67, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -73.37, "left": -53.5, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v33.24", "h-113.11", "v177.33" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "green", [ "v28.57", "h151.66", "v113.33" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "v15.24", "h164.44", "v112.67" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v41.91", "h167.22", "v156" ] ]
  ]
}
```

CIRCUIT DIAGRAM:



WOKWI LINK:

https://wokwi.com/projects/346919437834650194

WOKWI OUTPUT:

```
Connecting to ....
WiFi connected
IP address:
10.10.0.2
Reconnecting client to rv07c6.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance (cm): 188.00
Distance (cm): 31.00
ALERT!!
Sending payload: {"Distance":31.00,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 80.00
ALERT!!
Sending payload: {"Distance":80.00,"ALERT!!":"Distance less than 100cms"}
Publish ok
```

```
Distance (cm): 54.00
ALERT!!
Sending payload: {"Distance":54.00,"ALERT!!":"Distance less than
100cms"}
Publish ok
Distance (cm): 190.00
Distance (cm): 53.00
ALERT!!
Sending payload: {"Distance":53.00,"ALERT!!":"Distance less than
100cms"}
Publish ok
Distance (cm): 4.00
ALERT!!
Sending payload: {"Distance":4.00,"ALERT!!":"Distance less than
100cms"}
Publish ok
Distance (cm): 81.00
ALERT!!
Sending payload: {"Distance":81.00,"ALERT!!":"Distance less than
```

IBM CLOUD OUTPUT: