Assignment -4
Wokwi& IBM Cloud

Assignment Date	28 October 2022
Student Name	Kowsalya E
Student Roll Number	732219CS056
Maximum Marks	2 Marks

Question-1:

Write code and connections in wokwi for ultrasonic sensor. Whenever the distance is less than 100 cms sent "alert" to ibm cloud and display in device recent events.

Solution:

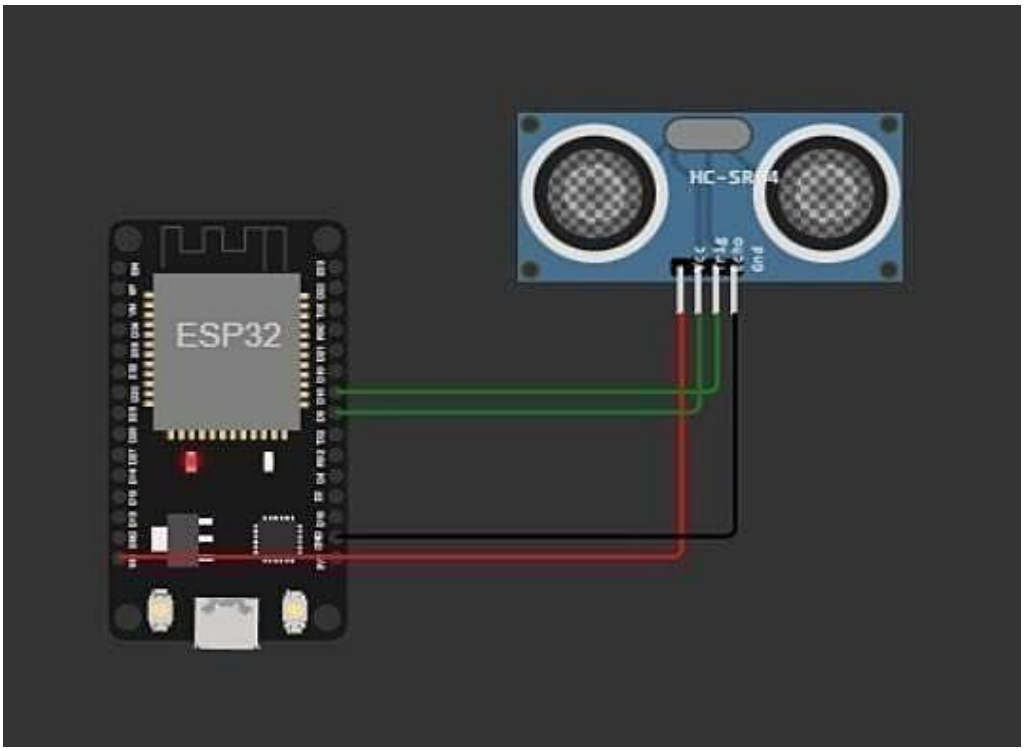
Code

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "zalrlt"//IBM ORGANITION ID
#define DEVICE_TYPE "iot"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "device113"//Device ID mentioned in ibm watson IOT Platfo
#define TOKEN "12345678" //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();
}
```

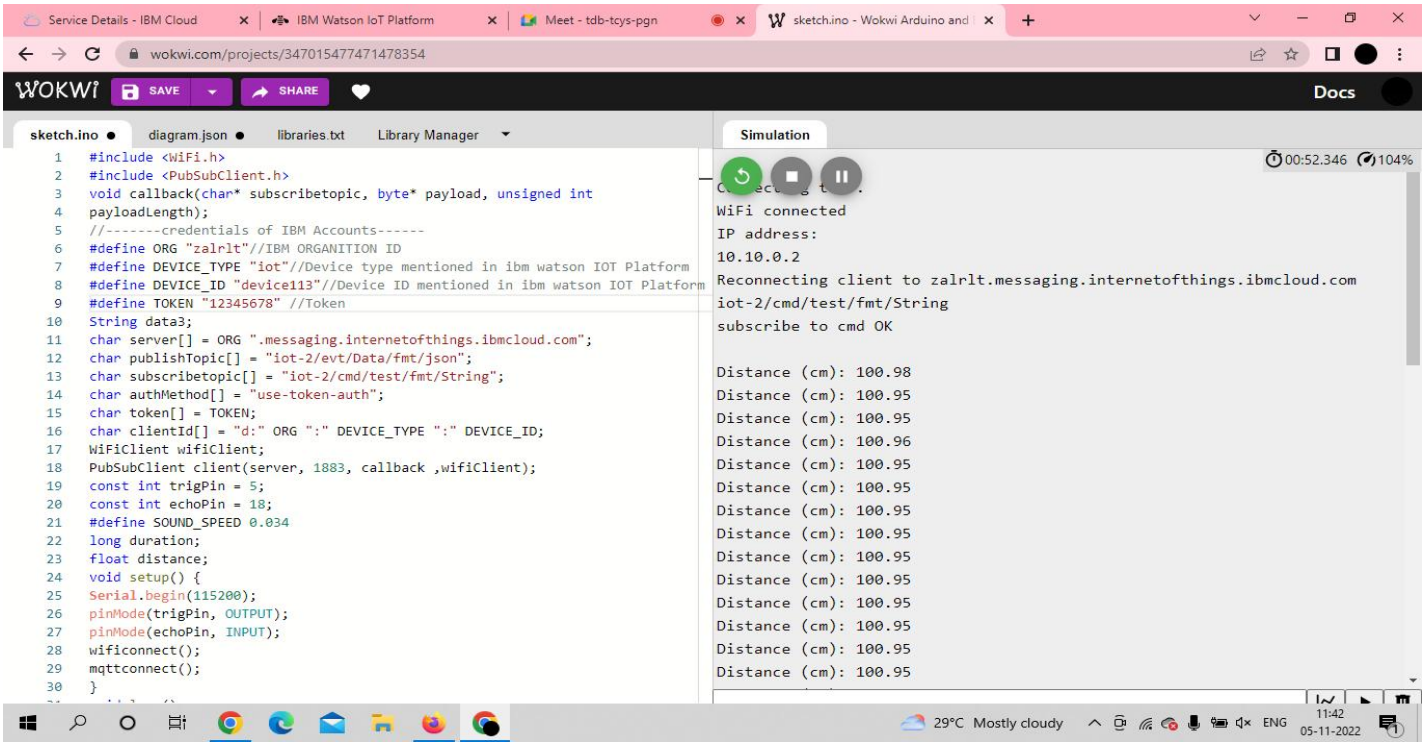
```
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
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 ");
```

```
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 {
100  Serial.println("subscribe to cmd FAILED");
101  }
102  }
103  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength
104  {
105  Serial.print("callback invoked for topic: ");
106  Serial.println(subscribetopic);
107  for (int i = 0; i < payloadLength; i++) {
108  //Serial.print((char)payload[i]);
109  data3 += (char)payload[i];
110  }
111  Serial.println("data: "+ data3);
112  data3="";
113  }
```

Connections:



Output (wokwi):



Link: https://wokwi.com/projects/347015477471478354

Output (IBM Cloud):