Assignment - 4

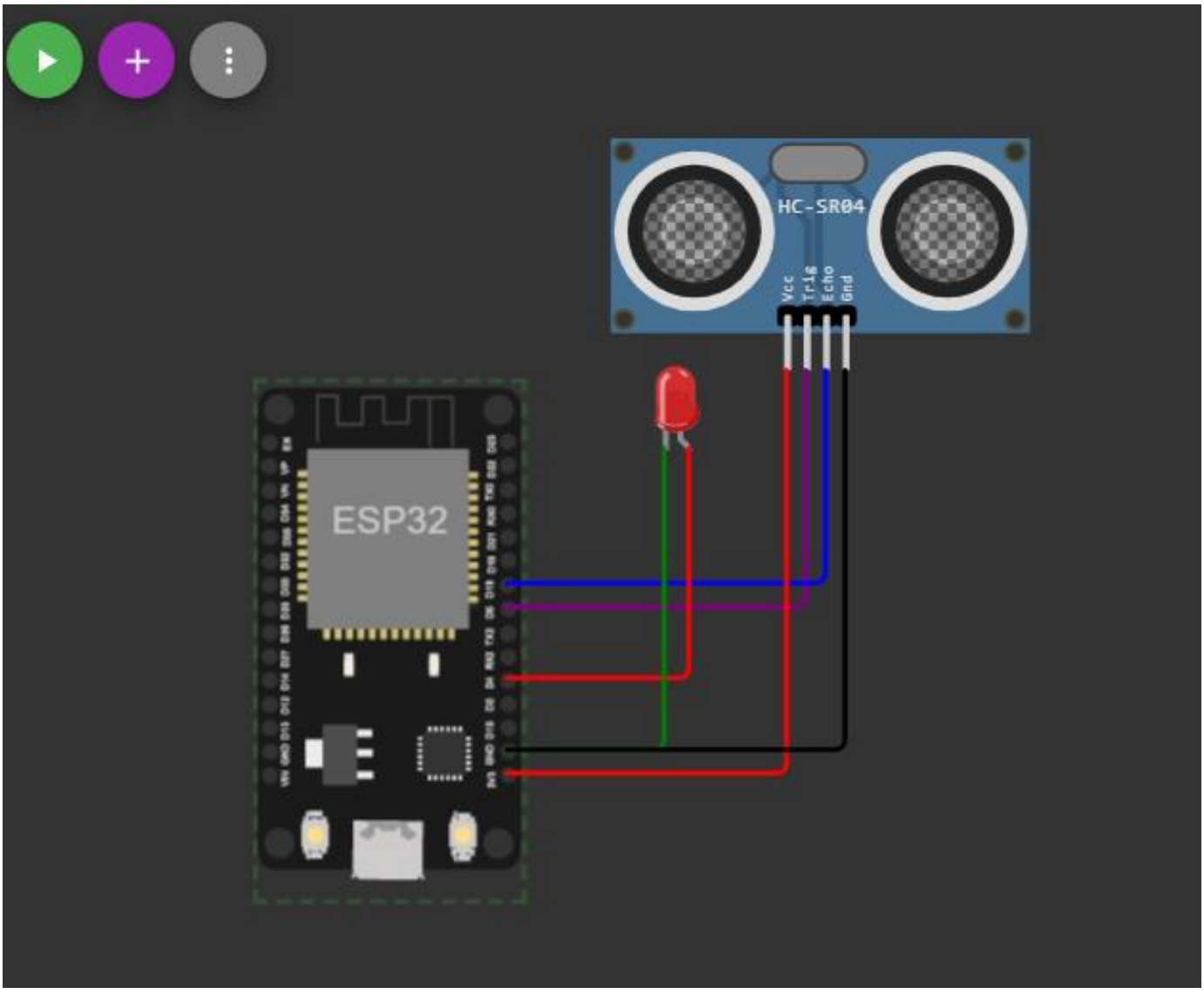
Assignment Date	18/10/2022
Student Name	THARUN PRASATH R C
Student Roll Number	61071912150
Maximum Marks	2 Marks

Question:

Write code and connections in WokWi for Ultrasonic Sensor. Whenever the distance is less 100cm, send “Alert” to IBM cloud and display in device recent events.

Solution:

Circuit Diagram:



Code:

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

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts-----

#define ORG "um5y3e"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "ESP3240P"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "Sv*Ygwum-RMJXiOBy?"//Token
String data3;
float dist;

//-----Customise the above values-------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fat/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientid[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


WiFiClient wificlient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback,wificlient); //calling the predefined client id by passing parameter like server id, portand wificredential

int LED = 4;
int trig = 5;
int echo = 18;
void setup()
{
  Serial.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED, OUTPUT);
  delay(10);
  wificonnect();
```

```
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
  digitalWrite(trig, LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  float dur = pulseIn(echo,HIGH);
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distance in cm");
  Serial.println(dist);


  PublishData(dist);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}


//..retrieving to cloud...

void PublishData(float dist) {
  mqttconnect();//function call for connecting to ibm
  //Creating the string in the form of Json to update the data to ibm cloud

  String object;
  if (dist <100)
  {
    digitalWrite(LED, HIGH);
    Serial.println("object is near");
```

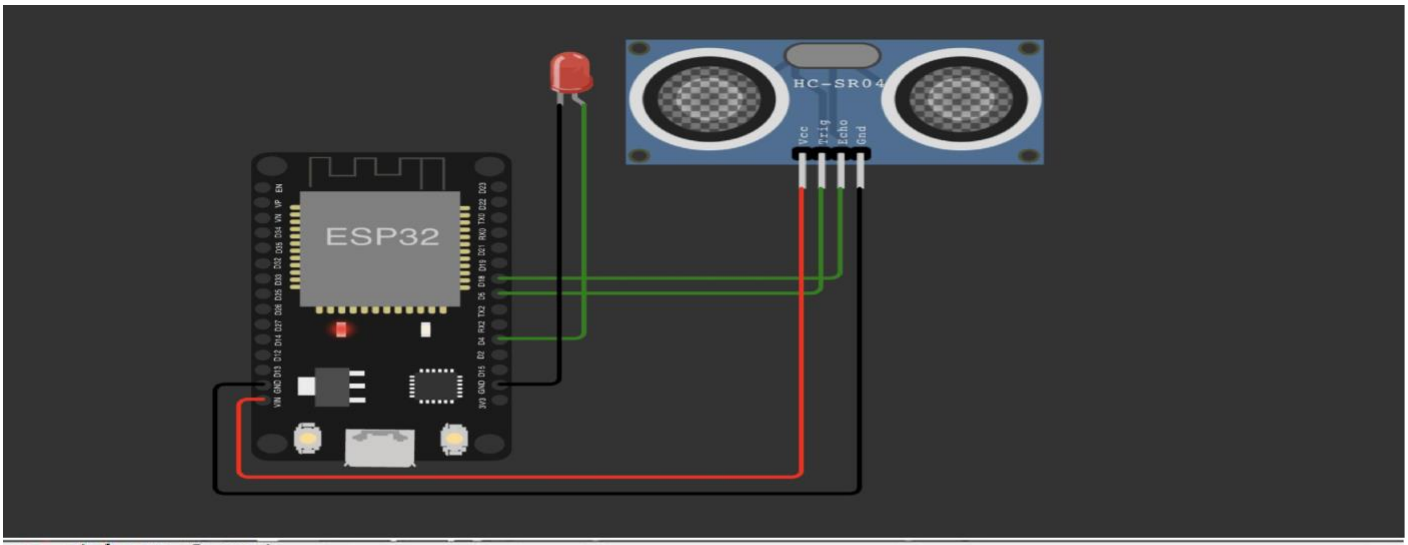
```
    Serial.println("object is near");
    object = "Near";
  }
  else
  {
    digitalWrite(LED, LOW);
    Serial.println("no object found");
    object = "No";
  }

  String payload="{\"distance\":";
  payload += dist;
  payload += "," "\"object\":\"";
  payload += object;
  payload += "\"}";

  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failm
  }
  else
  {
    Serial.println("Publish failed");
  }
}

void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.print(server);
    while (!!!client.connect(clientid, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
```

```
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("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[1]);
```

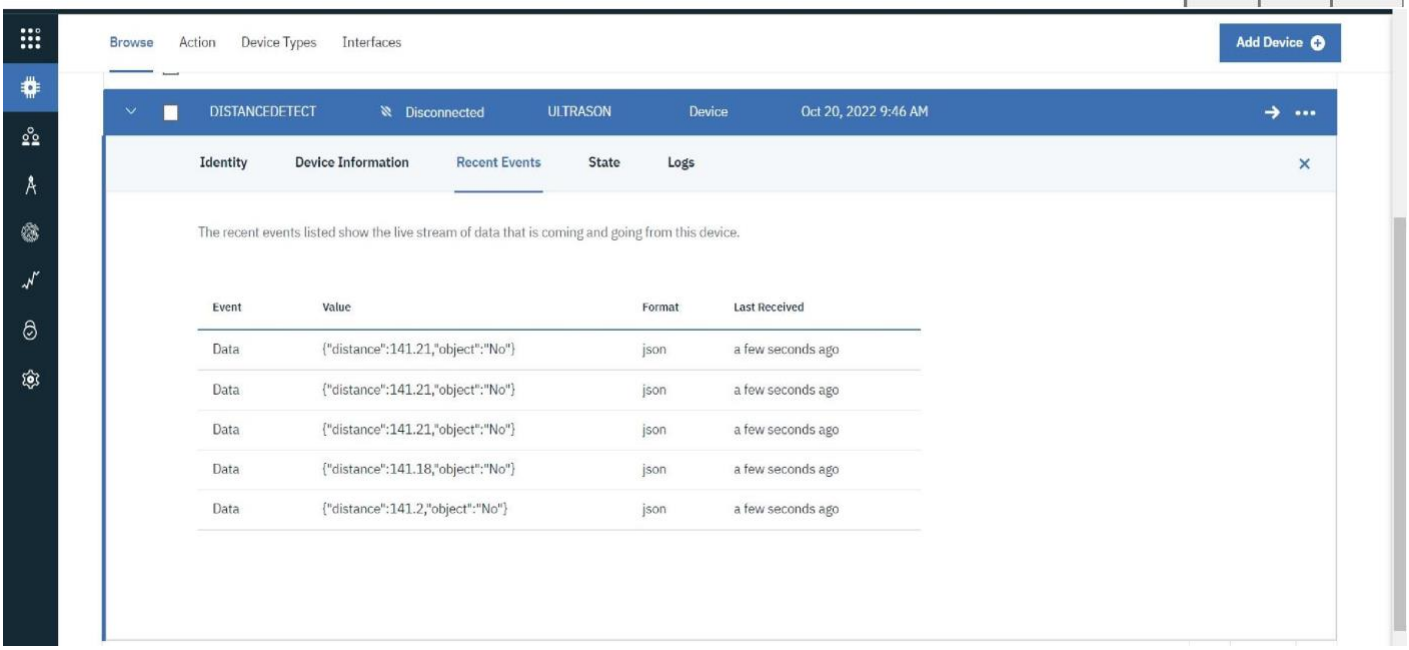
```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength){
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i=0; i < payloadLength; i++) {
    //Serial.print((char)payload[1]);
    data3 += (char)payload[1];
  }
  data3 = "";
}
```

OUTPUT:

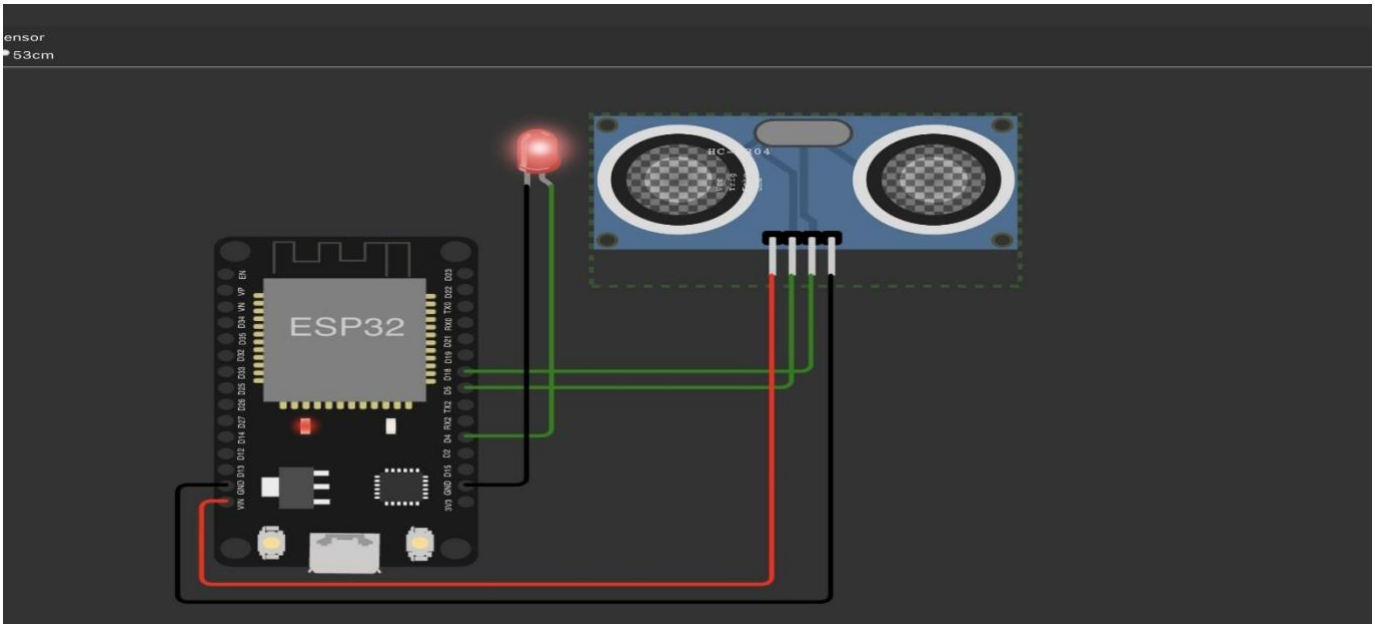
When the Object is Far:




```
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
Distancein cm141.21
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
```



When the Object is Near:




```
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
Distancein cm97.82
object is near
Sending payload: {"distance":97.82,"object":"Near"}
Publish ok
```