ASSIGNMENT 4

Date	18 October 2022
Team ID	PNT2022TMID19515
Project Name	Project – Real time river water quality monitoring and Control System
Maximum Marks	4 Marks

Project Title : Real-Time River Water Quality Monitoring And Control

Faculty Mentor : Mohanapriya A

Team ID: PNT2022TMID19515

Team Members:

1. Harish V - Team Leader
2. Nirmalkumar V S - Team Member
3. Mohammed Adhil H - Team Member
4. Jaisherma J Team - Member

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

ANSWER:

```
#include <WiFi.h>//librarv for wifi

#include <PubSubClient.h>//library for MQtt



void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

```
#define ORG "dymr4l"//IBM ORGANITION ID

#define DEVICE_TYPE "NodeMCU"

#define DEVICE_ID "2004" //Device ID mentioned in ibm watson IOT Platform

#define TOKEN "Nirmal@2002" //Token

String data3;
```

```
float dist;
```

```
char server[] = "dymr4l.messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/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();

mqttconnect();

}
```

```
void loop()// Recursive Function

{

digitalWrite(trig, LOW) ;

digitalWrite(trig,HIGH);

delayMicroseconds(10);

digitalWrite(trig, LOW) ;

float duration = pulseIn(echo,HIGH) ;

float distance = (duration * 0.0343)/2;

Serial.print("Distancein cm");

Serial.println(distance);
```

```
PublishData(distance);

delay(1000);

if(!client.loop()) {

mqttconnect();

}

}
```

```
void PublishData(float distance)

{

mqttconnect();//function call for

String object;

if(distance < 100)

{

digitalWrite(LED,HIGH);

Serial.println("object is near");
```

```
object = "Near";

}

else

{

digitalWrite(LED, LOW) ;

Serial.println("no object found");

object = "No";

}
```

```
String payload = "{\"distance\":";

payload += distance;

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 failed

} 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() //function defination for wificonnect
{
Serial.println();
Serial.print("Connecting to ");
```

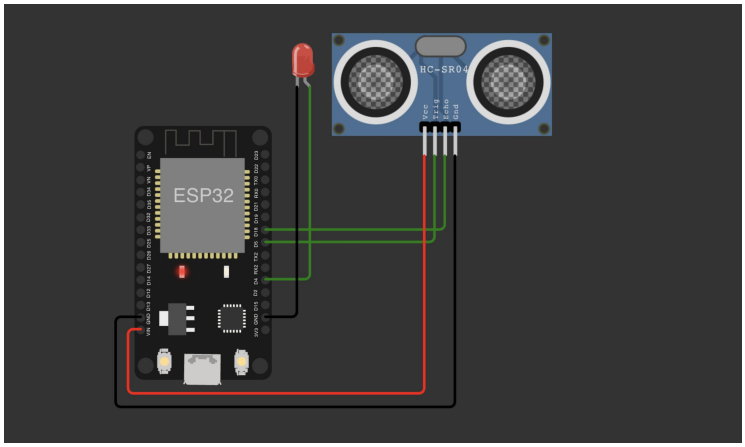
```
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
```

```
}
```

```
void initManagedDevice() {

if (client.subscribe(subscribetopic)) {

Serial.println((subscribetopic));

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}
```

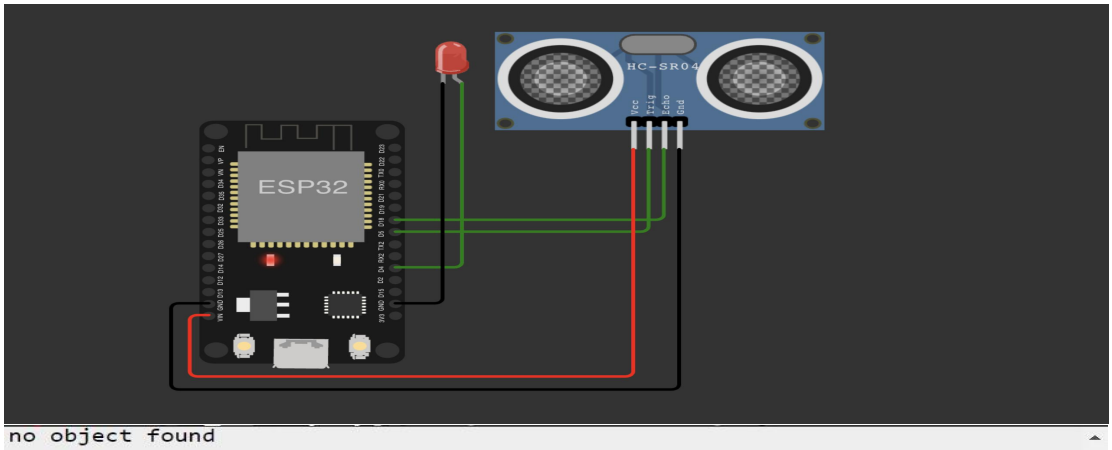
```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)

{

Serial.print("callback invoked for topic: ");

Serial.println(subscribetopic);

for (int i = 0; i < payloadLength; i++) {

data3 += (char)payload[i];

}

Serial.println("data: "+ data3);

data3="";

}
```

CIRCUIT:

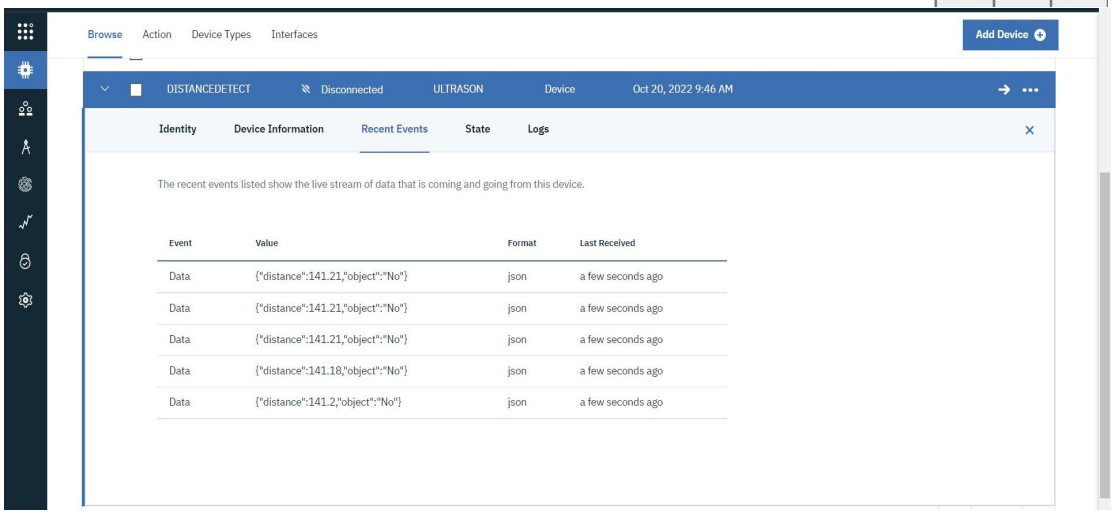


OUTPUT:

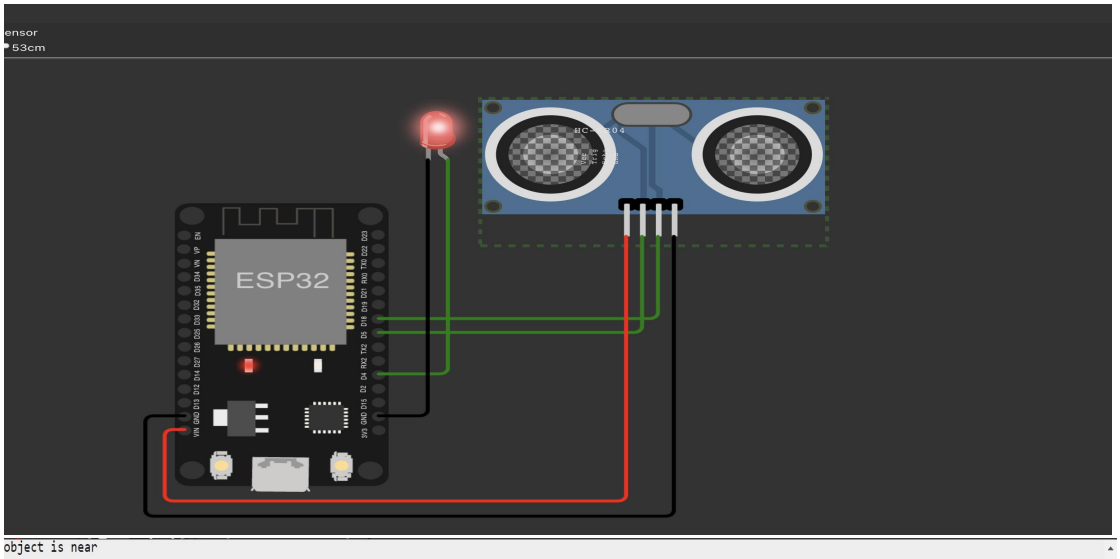
When object is near:-




```
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 object is far:-




```
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
```