ASSIGNMENT 4

Date	**25 October 2022**
Team ID	**PNT2022TMID43374**
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 system

Team ID: PNT2022TMID43374

Team Members:

1. **Kaviya P- Team Leader**
2. **Preethi T -Team Member**
3. **Praneetha S S- Team Member**
4. **Ramanya U S – Team Memberr**

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.

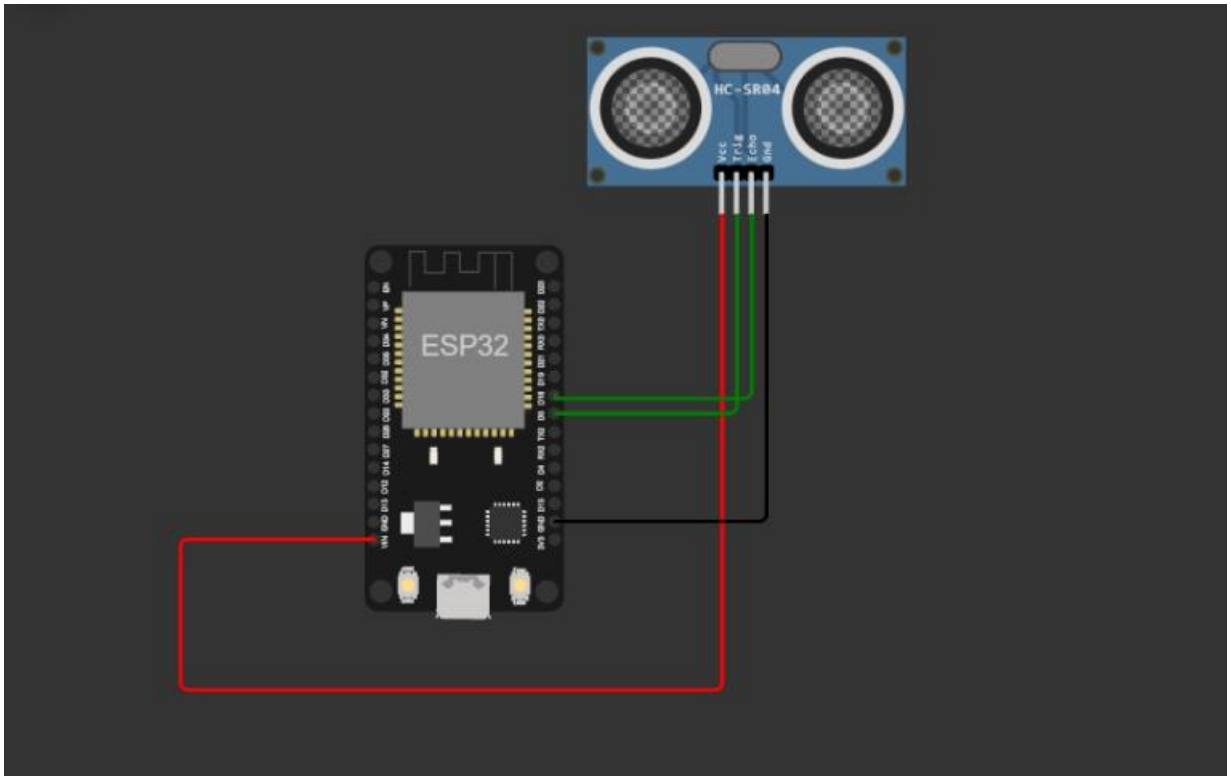
CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "9fds9f"
#define DEVICE_TYPE "kpr123"
#define DEVICE_ID "1122"
#define TOKEN "+_HOJ)L?Vl4OV8ZMm("
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/event_1/fmt/json ";
char topic[] = "iot-2/cmd/Test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

PubSubClient client(server, 1883, wifiClient);
void publishData();

const int trigpin=5;

const int echopin=19;
String command;
```

```
String data="";
String name="Alert";
 String icon="";
long duration;
int dist;
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
 pinMode(echopin, INPUT);
 wifiConnect();
 mqttConnect();
}
void loop() {
  publishData();
   delay(500);
if (!client.loop()) {
  mqttConnect();
}
}

void wifiConnect() {
Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{ delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
 Serial.println(WiFi.localIP());
}
void mqttConnect() {
if (!client.connected()) {
Serial.print("Reconnecting MQTT client to ");
 Serial.println(server);
 while (!client.connect(clientId, authMethod, token))
 { Serial.print(".");
Serial.print("*");
 delay(1000);
}
initManagedDevice();
Serial.println();
}
}

void initManagedDevice() {

if (client.subscribe(topic)) {
  Serial.println(client.subscribe(topic));
  Serial.println("subscribe to cmd OK");
}
```

```
else {
Serial.println("subscribe to cmd FAILED");
}
}

void publishData()
{
digitalWrite(trigpin,LOW); digitalWrite(trigpin,HIGH); delayMicroseconds(10);
digitalWrite(trigpin,LOW); duration=pulseIn(echopin,HIGH); dist=duration*speed/2;
if(dist<100){
dist=100-dist; icon="Not-Crashed";
}
else{ dist=0;
icon="Crashed";
}
DynamicJsonDocument doc(1024); String payload; doc["Name"]=name; doc["Impact"]=icon;
doc["Distance"]=dist; serializeJson(doc, payload); delay(3000);
Serial.print("\n"); Serial.print("Sending payload: "); Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) { Serial.println("Publish
OK");
}
else {
Serial.println("Publish FAILED");
}
}
```

SCHEMATIC/CIRCUIT DIAGRAM:



IBM CLOUD OUTPUT:

Event	Value	Format	Last Received
event_1	{"Name":"Alert","Impact":"Not-Crashed","Distan...	json	a few seconds ago
event_1	{"Name":"Alert","Impact":"Not-Crashed","Distan...	json	a few seconds ago
event_1	{"randomNumber":36,"temp":67,"hum":91}	json	a few seconds ago
event_1	{"Name":"Alert","Impact":"Not-Crashed","Distan...	json	a few seconds ago
event_1	{"Name":"Alert","Impact":"Not-Crashed","Distan...	json	a few seconds ago

WOKWI LINK:

https://wokwi.com/projects/347238295730651732