Assignment -4

TEAM ID: PNT2022TMID54475

Write code and connections in wokwi for ultrasonic sensor.Whenever distance is less than 100 cm send “Alert” to IBM cloud and display in device recent events. Upload document with wokwi share link and images of IBM cloud.

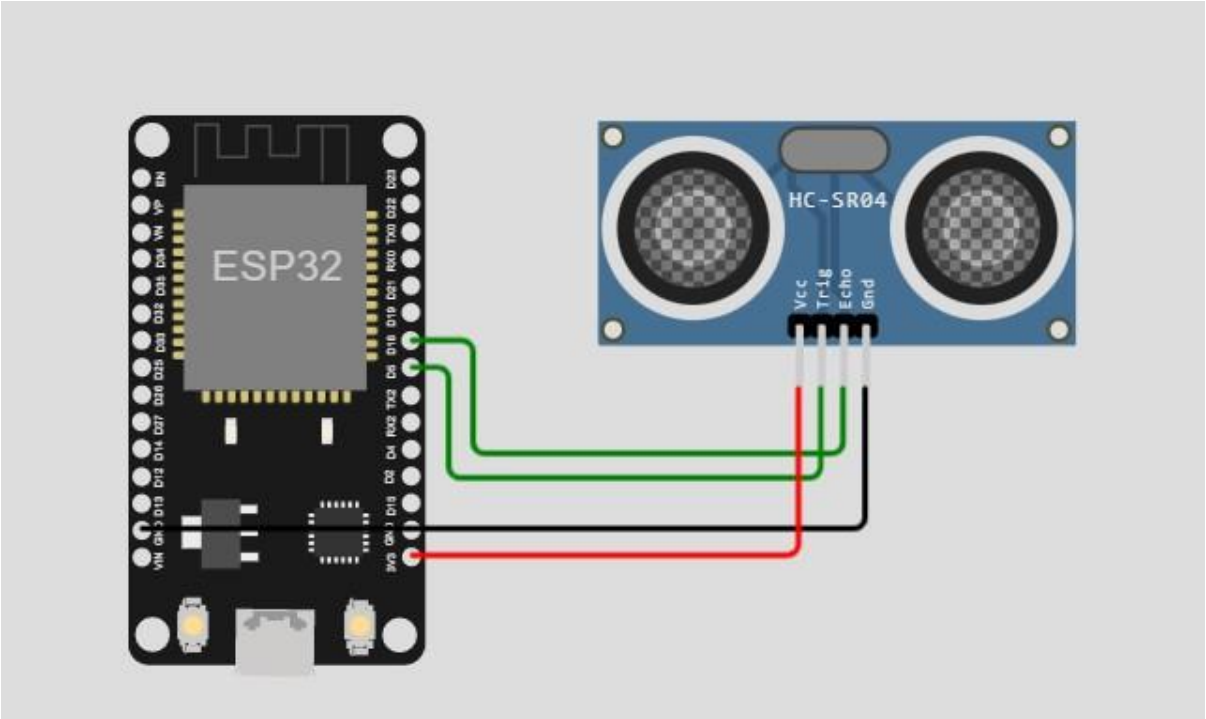
Code :

```
#include <WiFi.h>
#include<PubSubClient.h>
#include <ArduinoJson.h>
 void callback(char* subscribetopic,byte* payload,unsigned int
payloadLength); #define ORG "vy6h4o"
#define DEVICE_TYPE "akha215"
#define DEVICE_ID "akha215"
#define TOKEN "S7d2t()HWfU5eiM!vM"
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701
 const int trigPin = 5;
const int echoPin = 18;
 long duration;
float distanceCm;
float distanceInch;
String data;
 char server[]=ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/event_1/fmt/json"; char
subscribeTopic[] = "iot-2/cmd/home/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);

void setup() {
  Serial.begin(115200); // Starts the serial communication
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
wificonnect();   mqttconnect();
}
void loop() {   // Clears the
trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);   delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

```
  // Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

  // Calculate the distance
  distanceCm = duration * SOUND_SPEED/2;

  // Convert to inches   distanceInch =
distanceCm * CM_TO_INCH;

  // Prints the distance in the Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);

  delay(1000);

  PublishData(distanceCm);
delay(1000);   if(!client.loop())
  {
mqttconnect();
  }
}
void PublishData(float distanceCm)
{
  mqttconnect ();
String payload;

  if(distanceCm<100.0)
  {
    payload = "{\"Alert\":";
payload += distanceCm;     payload
+= "}";
  }
else   {
payload
=
"{\"dist
anceCm\"
:";
payload
+=
distance
Cm;
payload
+= "}";
  }



  Serial.print("Sending payload: ");
```

```
  Serial.print(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);

}
```




```
Distance (cm): 125.95
Sending payload: {"distanceCm":125.95}Publish ok
Distance (cm): 65.98
Sending payload: {"Alert":65.98}Publish ok
Distance (cm): 65.98
Sending payload: {"Alert":65.98}Publish ok
Distance (cm): 65.98
```

Event	Value	Format	Last Received
event_1	{"Alert":65.98}	json	a few seconds ago
event_1	{"Alert":65.98}	json	a few seconds ago
event_1	{"distanceCm":125.95}	json	a few seconds ago
event_1	{"distanceCm":125.95}	json	a few seconds ago
event_1	{"distanceCm":126}	json	a few seconds ago