

## CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient;

#define ORG "nhpwjc"
#define DEVICE_TYPE "NodeMCU"
#define DEVICE_ID "USE YOUR ID"
#define TOKEN "USE YOUR TOKEN"
#define speed 0.034
  char server[] =
ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/Data/fmt/json"; char topic[] =
"iot-2/cmd/home/fmt/String"; char authMethod[] = "usetoken-
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=18;
String command;
String data="";
  long
duration; float
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("."); delay(500);
        } 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);

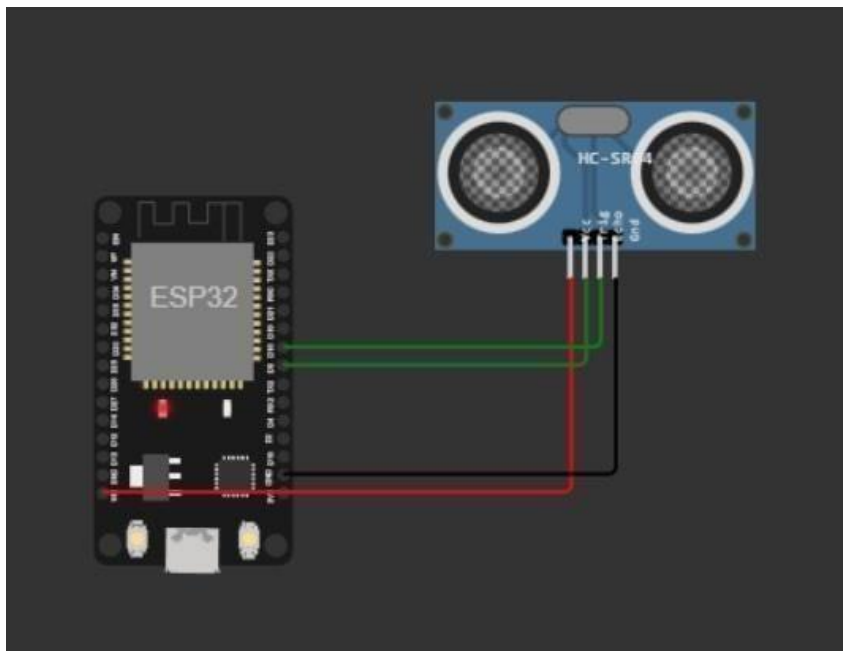
```

```

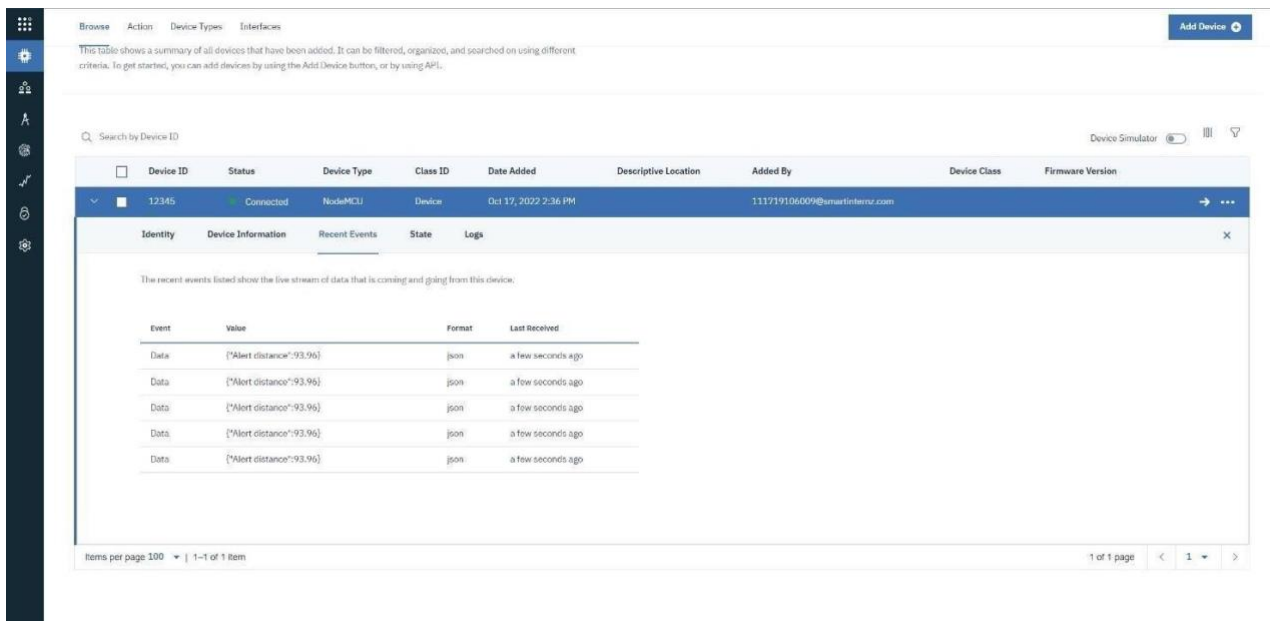
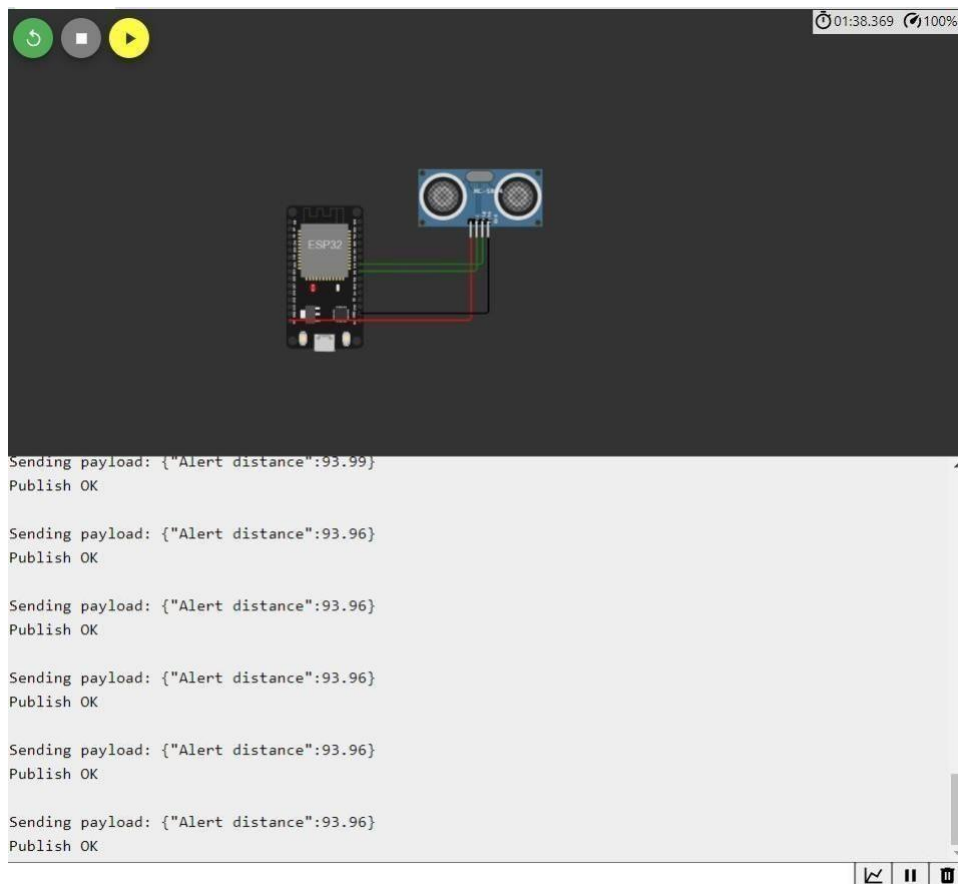
delayMicroseconds(10); digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH); dist=duration*speed/2;
if(dist<100){
  String payload = "{\"Alert distance\":\""; payload += dist; payload
  += "\"}";
  Serial.print("\n");
  Serial.print("Sending payload: "); Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str()))
    Serial.println("Publish OK");
  } else {
    Serial.println("Publish FAILED"); } }
}

```

## CONNECTIONS:



## OUTPUT:



**WOKWI LINK -**

<https://wokwi.com/projects/346405970317935188>