CODE AND CONNECTIONS FOR ULTRASONIC SENSOR IN WOKWI:

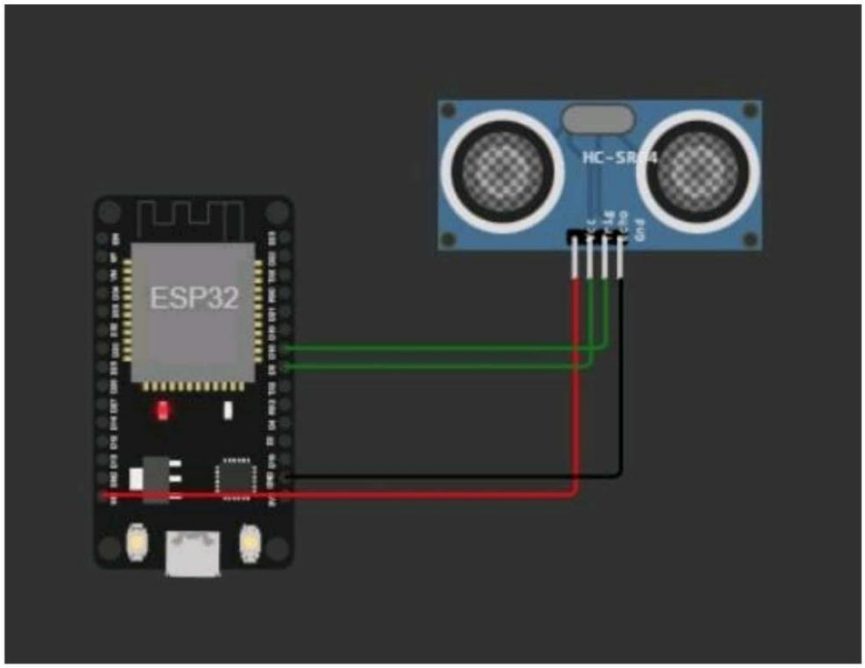
PROGRAM:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "lbklkq"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "rasp"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd_1/fmt/json”;
char topic[] = "iot-2/cmd/home/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=18;
String command;
String data=””;
String lat=”14.167589”;
String lon=”80.248510”;
String name=”point2”;
```

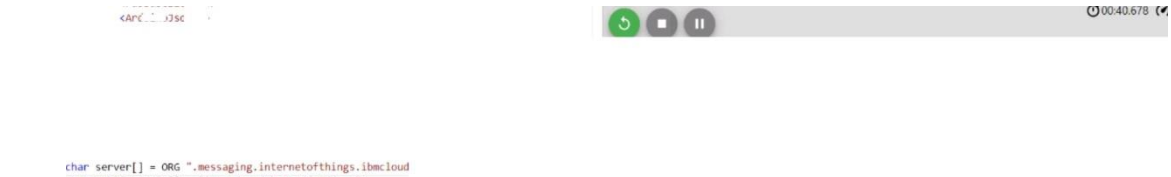
```
String icon=“”;

long duration;
int dist;

void setup()

        .     (115200);
         (trigpin, OUTPUT);
         (echopin, INPUT);
  wifiConnect();
  mqttConnect();


void loop() {

  publishData();
       (500);

  if (!client.loop()) {
    mqttConnect();



void wifiConnect() {
    :    .     (“Connecting to “);-            (“Wifi“);
  WiFi.   ’  (“Wokwi-GUEST“, “", 6);
  while (WiFi.status() != WL_C0NNECTED) {
         (500);
    Serial.print(".");

         .     (“WiFi connected, IP address: “);
       . ’     (WiFi.localIP());


void mqttConnect() {
  if (!client.connected()) {
                ("Reconnecting MQTT client to “); \.     .       (server);
    while (!client.connect(clientId, authMethod, token)) {
                  (".");
           (1000);

    initManagedDevice();
                  () :



void initManagedDevice() {
```

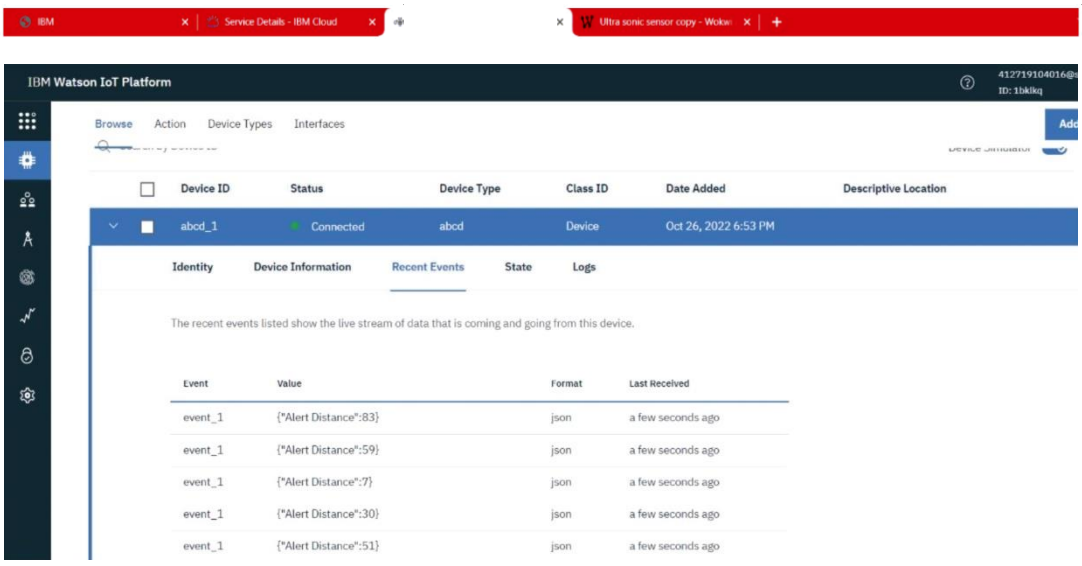
```
  if (client.subscribe(topic)) {
                  (client.subscribe(topic));
                 (“subscribe to cmd OK“);
  } else {
          .       (“subscribe to cmd FAILED“);


void publishData()

              (trigpin,L0W);
              (trigpin,HIGH);
                   (10);
              (trigpin,L0W);
  duration=.      (echopin,HIGH);
  dist=duration*speed/2;

  if(dist<100){
    dist=100-dist;
    icon=“fa-trash“;
  }else{
    dist=0;
    icon=“fa-trash-o“;

  DynamicJsonDocument doc(1024);
  String payload;
  doc[“Name“]=name;
  doc[“Latitude"]=lat;
  doc[“Longitude“]=lon;
  doc[“Icon“]=icon;
  doc[“FillPercent”]=dist;
  serializeJson(doc, payload);
       (3000);
              (“\n“);
              (“Sending payload: “);
       .        (payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
                  ("Publish OK");
  } else {
    Serial.r      (“Publish FAILED“);
```

CONNECTIONS:


HC-SR04
ESP32

OUTPUT:


00:40.678
char server[] = ORG ".messaging.internetofthings.ibmcloud
char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
char topic[] = "iot-2/cmd/home/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();
long duration;
int dist;
Publish OK
Sending payload:
{"Name":"point2","Latitude":"14.167589","Longitude":"80.248510","Icon":"fa-trash
o","FillPercent":0}


IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
abcd_1
Connected
abcd
Device
Oct 26, 2022 6:53 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
event_1
{"Alert Distance":83}
json
a few seconds ago
event_1
{"Alert Distance":59}
json
a few seconds ago
event_1
{"Alert Distance":7}
json
a few seconds ago
event_1
{"Alert Distance":30}
json
a few seconds ago
event_1
{"Alert Distance":51}
json
a few seconds ago