

ASSIGNMENT-4

MK.MOHAMED FARDEEN
421319106023(batch9)
Smartwastemanagementformetropolitancities

Question:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 centimeters it should send "alert" to IBM cloud and display in device recent events

Code:

```
#include<WiFi.h>
#include
<PubSubClient.h>#includ
e<ArduinoJson.h>
WiFiClient

wifiClient;#defineOR

G"9tg03j"

#defineDEVICE_TYPE"RaspberryPi"#define
DEVICE_ID"12345"
#defineTOKEN"12345678"
#define speed0.034

charserver[]="ORG".messaging.internetofthings.ibmcloud.com";charp
ublishTopic[]="iot-2/evt/status1/fmt/json";
chartopic[]="iot-
2/cmd/home/fmt/String";charauthMethod[]="
use-token-auth";
chartoken[]=TOKEN;
charclientId[]="d:"ORG":DEVICE_TYPE":DEVICE_ID;

PubSubClientclient(server,1883,wifiClient);vo
idpublishData();

constinttrigpin=5;
constintechopin=19;St
ring
command;Stringdata=
"";String
```

```

name="Alert";Stringic
on="";
long
duration;intdi
st;
voidsetup()
{
  Serial.begin(115200);pinMo
  de(trigpin,
  OUTPUT);pinMode(echopin
  ,
  INPUT);wifiConnect();mqtt
  Connect();
}
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("WiFiconnected,IPaddress:");Serial.println(WiFi.localIP());
}
voidmqttConnect(){
  if(!client.connected()){
    Serial.print("ReconnectingMQTTclientto");Serial.println(server);whi
    le (!client.connect(clientId, authMethod, token)) {Serial.print(".");
    Serial.print("*");
    delay(1000);
  }
  initManagedDevice();Se
  rial.println();
}
}

voidinitManagedDevice(){
if (client.subscribe(topic))
{Serial.println(client.subscribe(topic));Serial.pr
intln("subscribetocmdOK");
}
}

```

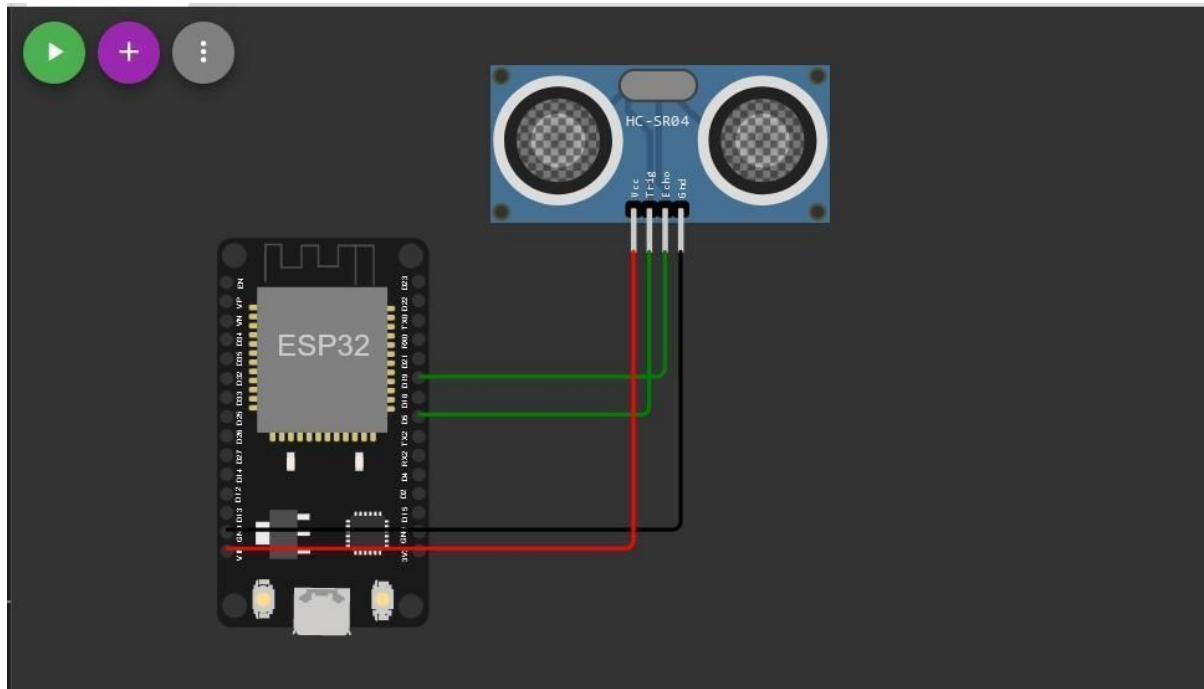
```

else{
    Serial.println("subscribetocmdFAILED");
}
}

voidpublishData()
{
    digitalWrite(trigpin,LOW);digital
    Write(trigpin,HIGH);delayMicrose
    conds(10);digitalWrite(trigpin,LO
    W);duration=pulseIn(echopin,HIG
    H);dist=duration*speed/2;if(dist<1
    00){
        dist=100-
        dist;icon="Not-
        Crashed";
    }
    else{ dis
        t=0;
        icon="Crashed";
    }
    DynamicJsonDocument
    doc(1024);String
    payload;doc["Name"]=name;doc["I
    mpact"]=icon;doc["Distance"]=dist
    ;serializeJson(doc,
    payload);delay(3000);
    Serial.print("\n");Serial.print("Sendi
    ngpayload:");Serial.println(payload);
    if(client.publish(publishTopic,(char*)payload.c_str())){ Serial.println
        ("PublishOK");
    }
    else{
        Serial.println("PublishFAILED");
    }
}

```

DIAGRAM:



OUTPUT:

```
esp32-blink.ino
34 Serial.begin(115200);
35 pinMode(trigpin, OUTPUT);
36 pinMode(echopin, INPUT);
37 wifiConnect();
38 mqttConnect();
39 }
40
41 void loop() {
42   publishData();
43   delay(500);
44   if (!client.loop()) {
45     mqttConnect();
46   }
47 }
48
49 void wifiConnect() {
50   Serial.print("Connecting to "); Serial.print("wifi");
51   WiFi.begin("Wokwi-GUEST", "", 6);
52   while (WiFi.status() != WL_CONNECTED) {
53     delay(500);
54     Serial.print(".");
55   }
56   Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
57 }
58
59 void mqttConnect() {
60   if (!client.connected()) {
61     Serial.print("Reconnecting MQTT client to "); Serial.println(server);
62     while (!client.connect(clientId, authMethod, token)) {
63       Serial.print(".");
64       Serial.print("a");
65       delay(1000);
66     }
67   }
68   initManagedDevice();
69 }
```

The screenshot shows the Arduino IDE interface with the 'esp32-arduino.ino copy' file open. The code on the left implements a loop that publishes data, connects to WiFi, and connects to an MQTT broker. The simulation window on the right displays the same hardware setup as the diagram above.

Add Device

ID	Name	Type	Status	Last Received	Device Location
12345	Rasp	Device	Connected	Oct 14, 2022 9:55 AM	

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
status1	{"Name":"Alert","Icon":"trash","FillPercent":0}	json	a few seconds ago
status1	{"Name":"Alert","Icon":"trash","FillPercent":0}	json	a few seconds ago
status1	{"Name":"Alert","Icon":"trash","FillPercent":0}	json	a few seconds ago
status1	{"Name":"Alert","Icon":"trash","FillPercent":0}	json	a few seconds ago

Items per page: 50 | 1–1 of 1 item

1 of 1 page < 1 >