

## ASSIGNMENT 4

**Assignment Date** :19 September 2022  
**Student Name** :Kathiravan R  
**Student Roll no** :622419104023  
**Maximum Marks** : 2 marks  
**Project Title** : IOT based safety gadget for child Monitoring& Notification

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cms send "alert" to ibm cloud and display in devicerecent events.

CODE:

```
#include
<WiFi.h>#include<PubSub
Client.h>
voidcallback(char*subscribetopic,byte*payload,unsignedintpayloa
dLength);
//-----credentialsofIBMAccounts-----
#defineORG"Ashfaq1824"//IBMORGANITIONID
#defineDEVICE_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform#define
e DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT
Platform#defineTOKEN"12345678"//Token
Stringdata3;
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";charpublishTopic[]="
"iot-2/evt/Data/fmt/json";
charsubscribetopic[]="iot-
2/cmd/test/fmt/String";charauthMethod[]="use-
token-auth";
chartoken[]=TOKEN;
charclientId[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;
WiFiClientwifiClient;
PubSubClient client(server, 1883,
callback ,wifiClient);const inttrigPin=5;
constintechoPin=18;
#defineSOUND_SPEED0.0341
ongduration;
float
distance;voids
etup(){
```

```

Serial.begin(115200);pinMode(trigPin,OUTPUT);pinMode(echoPin,INPUT);wificonnect();
mqttconnect();
}
voidloop()
{
digitalWrite(trigPin,LOW);delayMicroseconds(2);digitalWrite(trigPin,HIGH);delayMicroseconds(10);digitalWrite(trigPin,LOW);duration=pulseIn(echoPin,HIGH);
distance=duration*SOUND_SPEED/2;Serial.print("Distance(cm): ");Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);PublishData(distance);delay(1000);
if(!client.loop())
{mqttconnect();
}
}
delay(1000);
}
voidPublishData(floatdist)
{mqttconnect();
String payload = "{"+"Distance\":";payload+=dist;
payload
+=",\\"ALERT!!\":"+"\\Distancelessthan100cms\\";payload+="}";
Serial.print("Sendingpayload:");
Serial.println(payload);

if(client.publish(publishTopic,(char*)payload.c_str())){
Serial.println("Publishok");
}else{
Serial.println("Publishfailed");
}
}
voidmqttconnect(){
if(!client.connected())
{Serial.print("Reconnectingclientto");Serial.println(server);
while(!client.connect(clientId,authMethod,token)){
Serial.print(".");
delay(500);
}
}
}

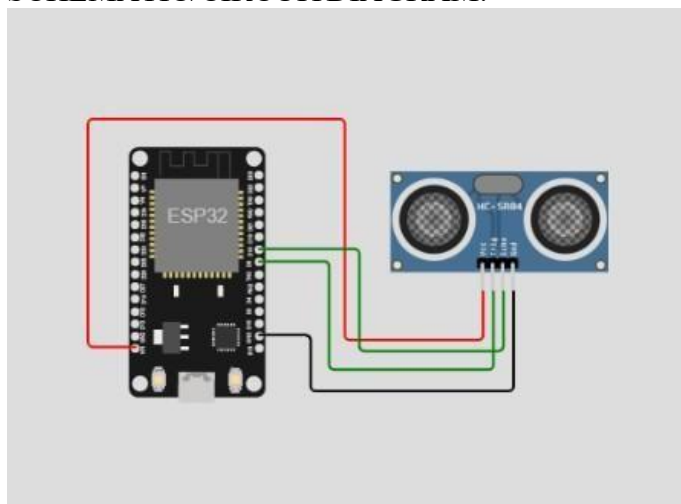
```

```

initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();Serial.print("Connect
ing to ");WiFi.begin("Wokwi-
GUEST","",6);
while(WiFi.status()!=WL_CONNECTED)
{delay(500);
Serial.print(".");
}
Serial.println("");Serial.println("
WiFiconnected");Serial.println("I
Paddress:
");Serial.println(WiFi.localIP());
}
void initManagedDevice(){
if (client.subscribe(subscribetopic))
{Serial.println((subscribetopic));Serial.println("su
bscribetocmdOK");
}else{
Serial.println("subscribetocmdFAILED");
}
}
void callback(char*subscribetopic,byte*payload,unsignedintpayloadLength)
{
Serial.print("callbackinvokedfortopic:");
Serial.println(subscribetopic);
for (inti=0; i<payloadLength;i++){
//Serial.print((char)payload[i]);
data3+=(char)payload[i];
}
Serial.println("data:"+data3);da
ta3="";
}

```

SCHEMATIC/CIRCUITDIAGRAM:



## IBM CLOUD OUTPUT:

The screenshot displays the IBM Cloud Output interface. On the left is a dark sidebar with icons for various services. The main content area has a top navigation bar with tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device' button with a plus icon is in the top right. Below the navigation bar is a sub-header with tabs: 'Identity', 'Device Information', 'Recent Events' (which is selected), 'State', and 'Logs'. A close button 'X' is in the top right of this sub-header. The main content area contains a text block stating: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this is a table with four columns: 'Event', 'Value', 'Format', and 'Last Received'. The table contains four rows of data, all with the event name 'event\_1' and a JSON value containing distance and alert information. The format for all rows is 'json', and the last received time for all is 'a few seconds ago'.

Event	Value	Format	Last Received
event_1	{"distance":7,"Alert":"Distance less than 10"}	json	a few seconds ago
event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago
event_1	{"distance":8,"Alert":"Distance less than 10"}	json	a few seconds ago
event_1	{"distance":9,"Alert":"Distance less than 10"}	json	a few seconds ago

## WOKWILINK:

<https://wokwi.com/projects/322410731508073042>