

# RAJALAKSHMI ENGINEERING COLLEGE

[Department of computer science and engineering]

## ASSIGNMENT -04

NAME: S HARISH KUMAR

TOPIC: Write a code and connections in wokwi for ultrasonic sensor

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h> void callback(char*
subscribetopic, byte* payload, unsigned int
payloadLength);

#define ORG "92zbfC"
#define DEVICE_TYPE "esp32"
#define DEVICE_ID "12345"
#define TOKEN "12345678" String data3; char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/Data/fmt/json"; char subscribetopic[]
= "iot-2/cmd/test/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); const int
trigPin = 5; const int echoPin = 18;
#define SOUND_SPEED 0.034 long
duration; float distance; void setup()
{ Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wifiConnect(); mqttConnect();
}
void loop() {
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)
; }
void PublishData(float dist) { mqttconnect();
String payload = "{"Distance\":";
payload += dist; payload +=
",\"ALERT!!\":"\"Distance less than
100cms\":"; payload += "}";
Serial.print("Sending payload: ");
Serial.println(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");

```

```

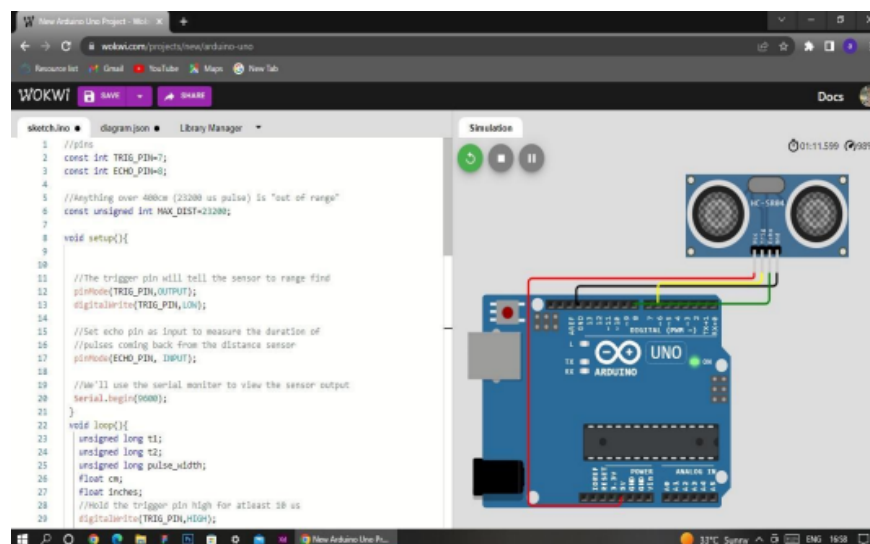
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) { Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}}
void callback(char* subscribetopic, byte* payload,
unsigned int payloadLength) {
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic); for (int i = 0; i
< payloadLength; i++)
{
data3 += (char)payload[i];
}
Serial.println("data: "+ data3); data3="";
}

```

WOKWI LINK :

<https://wokwi.com/projects/345395196387656275>

OUTPUT:



Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

The screenshot displays the IBM Watson IoT Platform interface. At the top, a 'Delete' button is visible. Below it, a table lists device details:

Device ID	Status	Device Type	Class ID	Date Added	Describe Location
22345	Connected	esp32	Device	Nov 1, 2021 9:53 PM	

Below the table, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is selected, showing a list of events:

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	["Distance":72.96,"ALDRS":["Distance less than ...	json	a few seconds ago
Data	["Distance":72.96,"ALDRS":["Distance less than ...	json	a few seconds ago
Data	["Distance":72.96,"ALDRS":["Distance less than ...	json	a few seconds ago

At the bottom, there is a status bar showing '2001', 'Disconnected', 'nephren.py', 'Device', and '0 Simulations running'.