

Assignment-4

Assignment Date	31.10.2022
Student Name	Aarthi P
Student Roll No	811019106001
Maximum Marks	2 Marks

Question :

Write code and connections in wokwi for ultrasonic sensor.

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

Solution :

Wokwi Link

<https://wokwi.com/projects/346957939801588308>

Code :

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

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

```
WiFiClient wifiClient;
```

```
String data3;
```

```
#define ORG "szro21"
```

```
#define DEVICE_TYPE "aarthidevicetype"
```

```
#define DEVICE_ID "aarthideviceid"
```

```
#define TOKEN "0987654321"
```

```
#define speed 0.034
```

```
#define led 15
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
```

```
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/command/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=13;
const int echopin=12;
String command;
String data="";
```

```
long duration;
float dist;
```

```
void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
```

```
wifiConnect();  
mqttConnect();  
}
```

```
void loop()  
{  
  bool isNearby = dist < 100;  
  digitalWrite(led, isNearby);
```

```
  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("IBM 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 += "\"}";

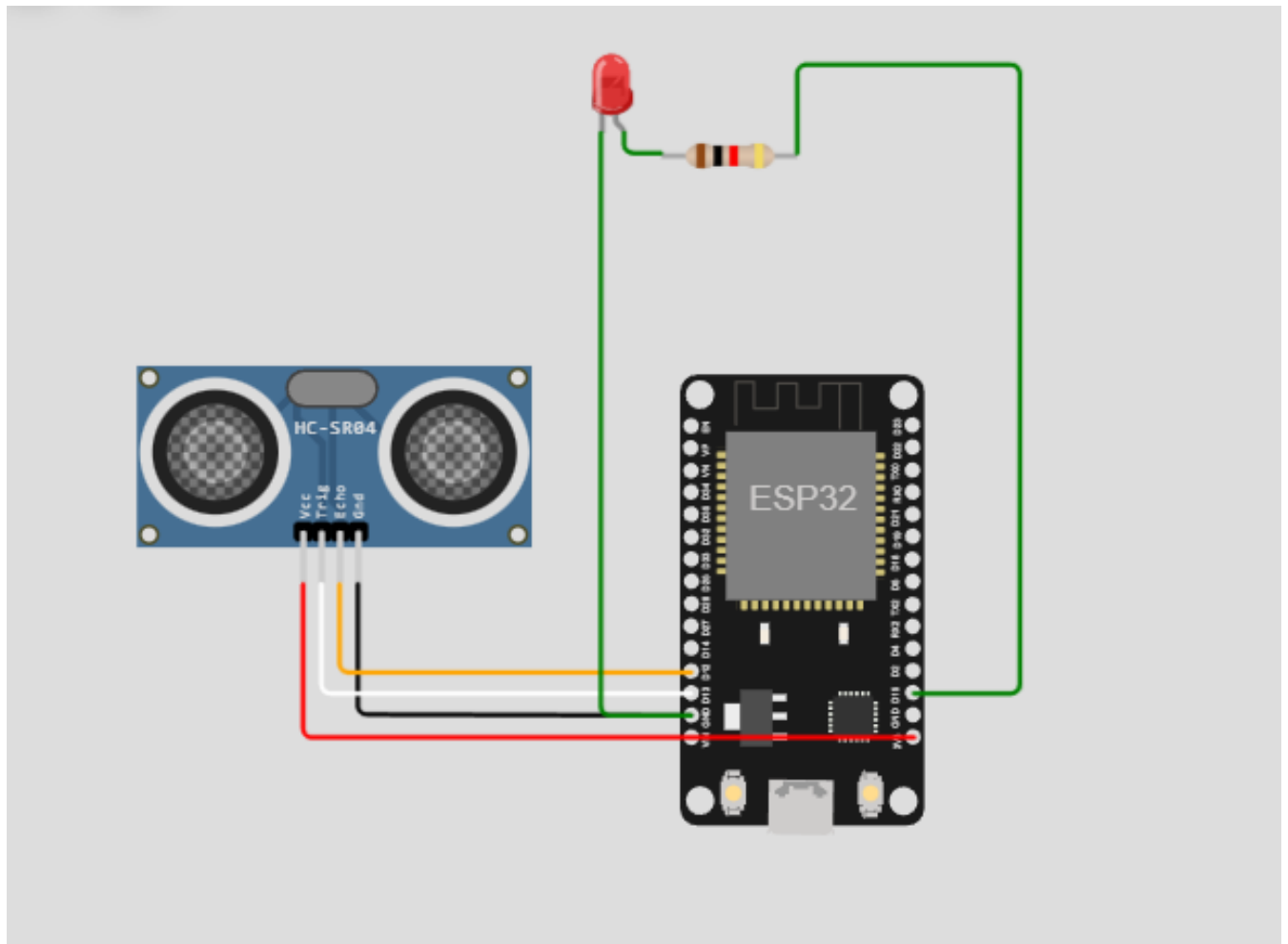
        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str()))

```

```
{  
  Serial.println("Publish OK");  
}  
  
}  
if(dist>101 && dist<150)  
{  
  String payload = "{\\\"Normal distance\\\":";  
  payload += dist;  
  payload += "}";  
  
  Serial.print("\\n");  
  Serial.print("Sending payload: ");  
  Serial.println(payload);  
  if(client.publish(publishTopic, (char*) payload.c_str()))  
  {  
    Serial.println("Warning crosses 110cm -- it automaticaly of the loop");  
    digitalWrite(led,HIGH);  
  }  
  else  
  {  
    Serial.println("Publish 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++)  
    {  
        dist += (char)payload[i];  
    }  
    Serial.println("data:" + data3);  
    if(data3=="lighton")  
    {  
        Serial.println(data3);  
        digitalWrite(led,HIGH);  
    }  
    data3="";  
}
```

Circuit Diagram :



Output :

The image displays two screenshots from a computer screen. The top screenshot shows the IBM Watson IoT Platform dashboard. The bottom screenshot shows the Wokwi simulation environment.

IBM Watson IoT Platform Dashboard:

- URL: `szro21.internetofthings.ibmcloud.com/dashboard/devices/browse`
- User: `811019106004@smartinfernz.com`, ID: `szro21`
- Navigation: Browse, Action, Device Types, Interfaces
- Device List:

Device ID	Status	Device Type	Device Name	Last Seen
123deviceid	Disconnected	ammudevicer	Device	Oct 12, 2022 12:25 PM
aarhideviceid	Connected	aarhidevicetype	Device	Oct 31, 2022 7:49 AM

- Selected Device: **aarhideviceid** (Connected)
- Recent Events:

Event	Value	Format	Last Received
Data	<code>{"Alert Distance":47.96}</code>	json	a few seconds ago
Data	<code>{"Alert Distance":47.96}</code>	json	a few seconds ago
Data	<code>{"Alert Distance":47.96}</code>	json	a few seconds ago
Data	<code>{"Alert Distance":47.96}</code>	json	a few seconds ago
Data	<code>{"Alert Distance":47.96}</code>	json	a few seconds ago

Wokwi Simulation:

- Project: `sketch.ino`
- Code Snippets:

```
124 {
125   String payload = "\\Normal distance\\:";
126   payload += dist;
127   payload += ";";
128
129   Serial.print("\n");
130   Serial.print("Sending payload: ");
131   Serial.println(payload);
132   if(client.publish(publishTopic, (char*) payload.c_str()))
133   {
134     Serial.println("Warning crosses 110cm -- it automatically of the loop");
135     digitalWrite(led,HIGH);
136   }
137   else
138   {
139     Serial.println("Publish FAILED");
140   }
141 }
142
143
144
145 }
146
147 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength)
148 {
149   Serial.print("callback invoked for topic:");
150   Serial.println(subscribeTopic);
151   for(int i=0; i<payloadLength; i++)
152   {
153     dist += (char)payload[i];
154   }
155   Serial.println("data:"+ data3);
156   if(data3=="lighton")
157   {
158     digitalWrite(led,HIGH);
159   }
160 }
```

- Simulation Status: 00:25.208, 88%
- Simulation Components: HC-SR04 sensor, ESP32 microcontroller, LED.
- Simulation Log:

```
Publish OK
Sending payload: {"Alert Distance":47.96}
Publish OK
Sending payload: {"Alert Distance":47.96}
Publish OK
```