GAS LEAKAGE MONITORING AND ALERTINGSYSTEM

Team ID: PNT2022TMID28798   kommi sai Venkata pavan kumar

ASSIGNMENT 4:

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

Upload document with wokwi share link and images of IBM cloud.

CODE:

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

#define ORG "6a4pz2"
#define DEVICE_TYPE "esp32"
#define DEVICE_ID "6281"
#define TOKEN "12345678"
#define trigpin 5
#define echopin 18
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
 char publishTopic[] = "iot-2/evt/data/fmt/json";
  char authMethod[] = "use-token-auth";
    char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
 long duration;
 float dist;
  void setup()
{
 Serial.begin(9900);

 pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
 }
void loop() {
  publishData();
  delay(500);
  if (!client.loop())
```

```cpp
{
 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);
 }
  Serial.println();
 }
  }
void publishData()
{  digitalWrite(trigpin,LOW);
 digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
   dist=(duration*0.034) /2;
    if(dist<100)
 {
 String payload = "{\"Distance\":";
  payload += dist;
   payload += ",";
    payload += "\"Status\":";
    payload += "\"Alert\"}";
 Serial.print("\n");
 Serial.print("Sending payload: ");
  Serial.println(payload);
   if (client.publish(publishTopic, (char*) payload.c_str())) {
```

```
Serial.println("Publish OK");
}
}
if(dist>100)
{
String payload = "{\"Distance\":";
payload += dist;
payload += ",";
payload += "\"Status\":";
payload += "\"Normal\"}";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
}
```

Diagram json:

```
{
  "version": 1,
  "author": "pavan kumar",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 38.67, "left": -88,
"attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -43.48, "left":
45.5, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "esp:VIN", "ultrasonic1:VCC", "red", [ "h0" ] ],
    [ "esp:GND.1", "ultrasonic1:GND", "black", [ "h0" ] ],
    [ "esp:D5", "ultrasonic1:TRIG", "green", [ "h0" ] ],
    [ "esp:D18", "ultrasonic1:ECHO", "green", [ "h0" ] ]
  ]
}
```

Wokwi link: https://wokwi.com/projects/347641329302372947
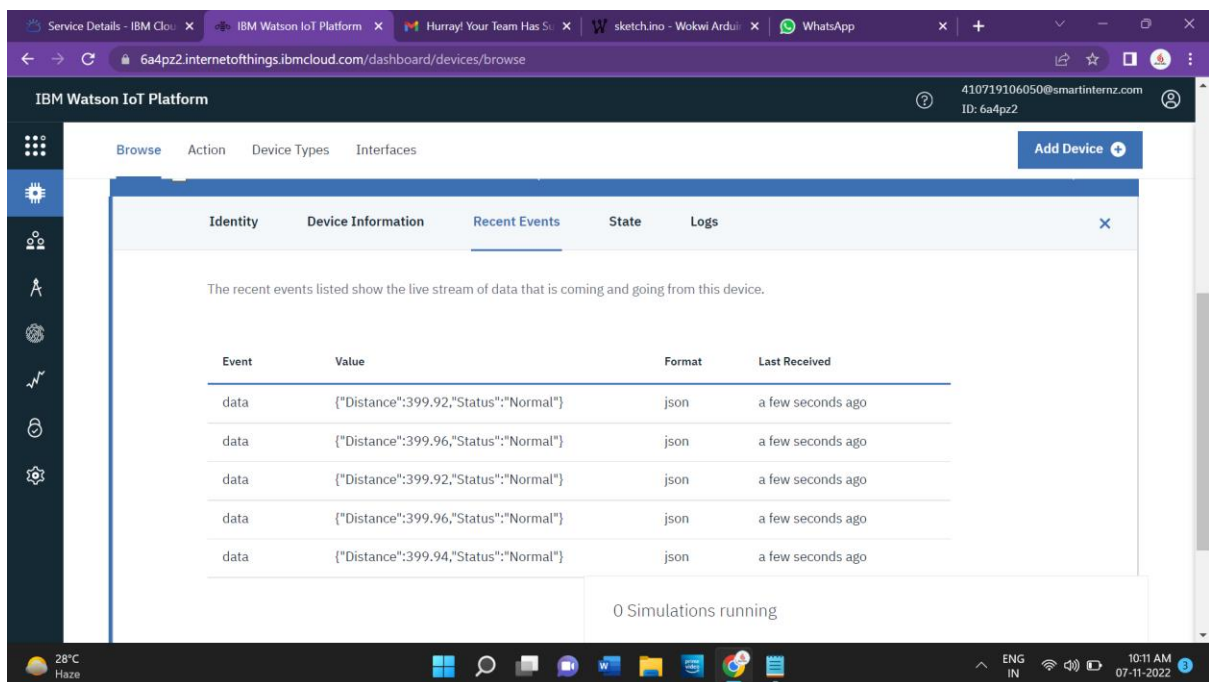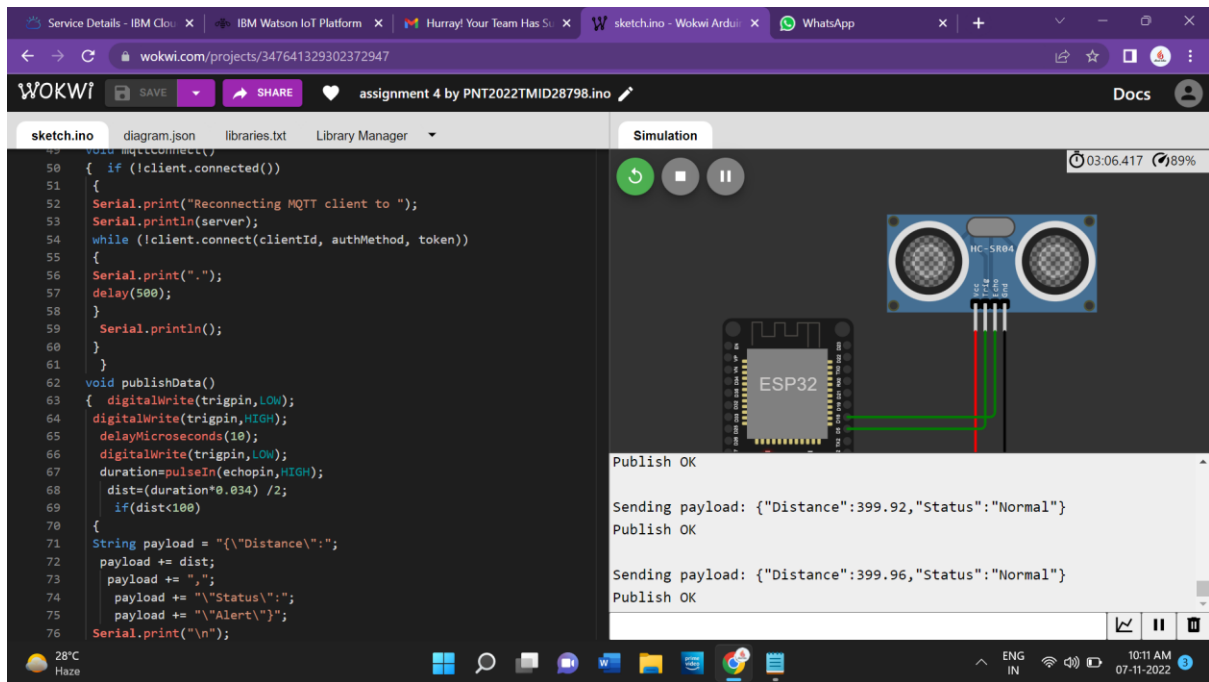
Assignment 4 by PNT2022TMID28798 team has completed successfully