

ASSIGNMENT- 4

Ultrasonic sensor simulation in Wokwi

Date	29 October2022
Team ID	PNT2022TMID28767
Project Name	IoT Enabled Smart Farming Application
Maximum Marks	2Marks

Question-1:

Write a code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms send an "Alert" to IBM cloud and display in the device recent events.

```
#include <WiFi.h>
#include <PubSubClient.h> void
callback(char* subscribetopic, byte*
payload, unsigned int payloadLength);
#define ORG "Lotq5"
#define DEVICE_TYPE "PSP32"
#define DEVICE_ID "WORKASSIGNMENT4"
#define TOKEN "345678" /
String data3;
charORG".messaging.internetofthings.ibmclou
d.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;
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
Serial.print("Reconnecting client to ");
Serial.println(server);

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);
}
}

```

```

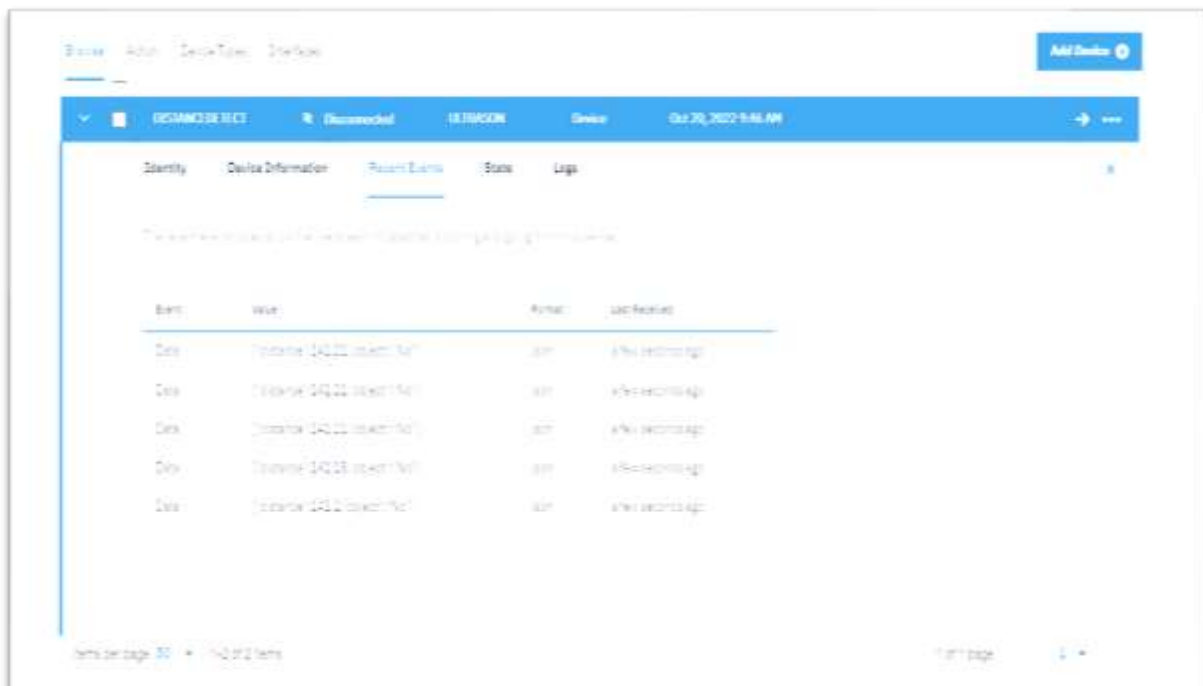
initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() !=WL_CONNECTED)
{ d
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
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++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}

```

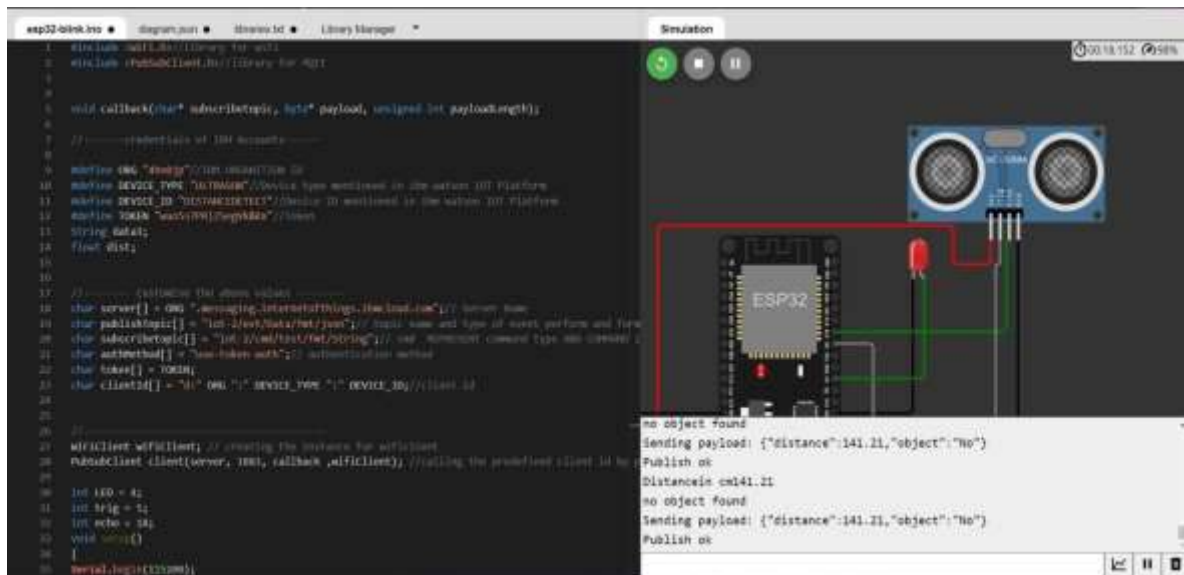
Diagram. json:

```
{
  "version": 1,
  "author": "Aravind",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -4.67, "left": -114.67, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": 15.96, "left": 89.17, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [
      "esp:VIN",
      "ultrasonic1:VCC",
      "red",
      [ "h-37.16", "v-178.79", "h200", "v173.33", "h100.67" ]
    ],
    [ "esp:GND.1", "ultrasonic1:GND", "black", [ "h39.87", "v44.04", "h170" ] ],
    [ "esp:D5", "ultrasonic1:TRIG", "green", [ "h54.54", "v85.07", "h130.67" ] ],
    [ "esp:D18", "ultrasonic1:ECHO", "green", [ "h77.87", "v80.01", "h110" ] ]
  ]
}
```

OUTPUT:



Data send to the IBM cloud device when the object is far:



```
1 #include <Arduino.h>
2 #include <PubSubClient.h>
3
4 void callback(char* topic, byte* payload, unsigned int payloadLength);
5
6 //----- Credentials of IoT Account -----
7
8 #define ONE "one" // IoT username ID
9 #define DEVICE_TYPE "ULTRASONIC" // Device type mentioned in the action IoT Platform
10 #define DEVICE_ID "DISTANCEDETECT" // Device ID mentioned in the action IoT Platform
11 #define TOKEN "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9" // Token
12
13 String data;
14 float dist;
15
16 //----- Customise the above values -----
17
18 char server[] = ONE; // Messaging.internetofthings.ibmcloud.com // Server Name
19 char publishTopic[] = "iot-1/act/data/hot/jan"; // Topic Name and type of event perform and fire
20 char subscribeTopic[] = "iot-1/act/data/hot/string"; // Use # character command type not allowed
21 char authMethod[] = "bearer auth"; // Authentication method
22 char token[] = TOKEN;
23 char clientId[] = "0" ONE "/" DEVICE_TYPE "/" DEVICE_ID "/" CLIENT_ID;
24
25 //-----
26
27 WiFiClient wifiClient; // creating the instance for wifiClient
28 PubSubClient client(server, 1883, callback, wifiClient); // calling the predefined client ID for
29
30 int LED = 4;
31 int trig = 5;
32 int echo = 14;
33 void setup()
34 {
35   Serial.begin(115200);
36 }
```

Simulation

no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
Distancein cm141.21
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok

Data sent to the IBM Cloud Device when the object is near:

