

Ultrasonic sensor simulation in Wokwi

TEAM ID	NM2023TMID10876
PROJECT NAME	Industrial Workers Health and Safety System Based on Internet of Things
Date	12 MAY 2023

Problem statement:

Build wokwi product, use ultrasonic sensor and detect the distance from the object. Whenever distance is less than 100cms upload the value to the ibm cloud.in recent device events upload the data from wokwi.

Wokwi link: <https://wokwi.com/projects/364770802247914497>

Code:

```
//Ultrasonic sensor with ibm cloud//

#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
#define ORG "3nfq88"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "1234"
#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 {

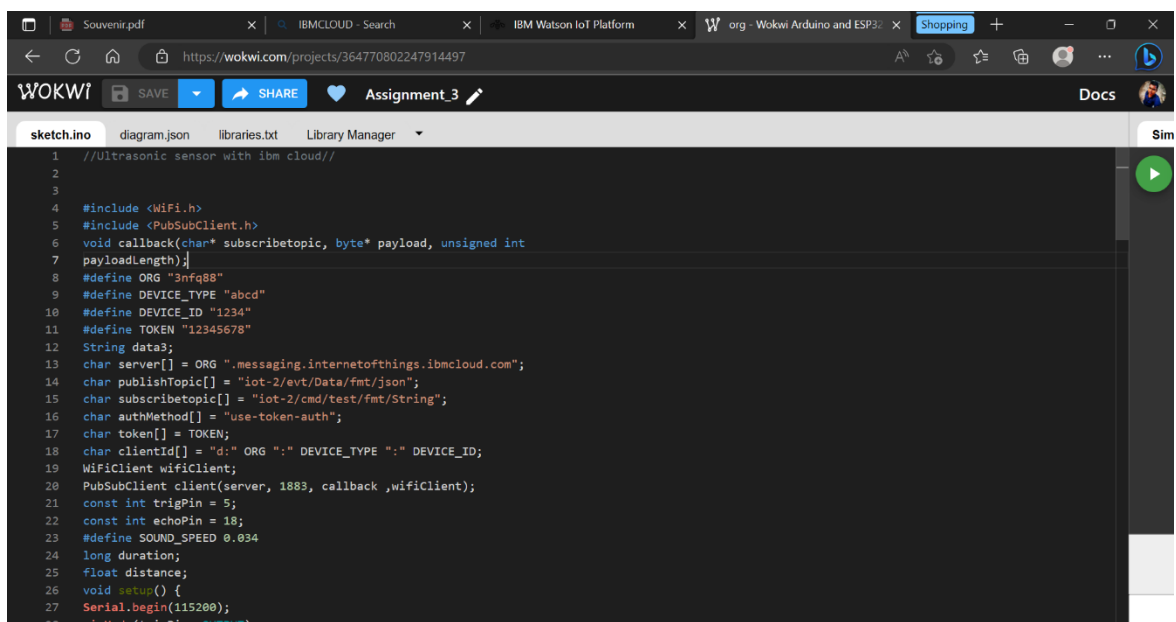
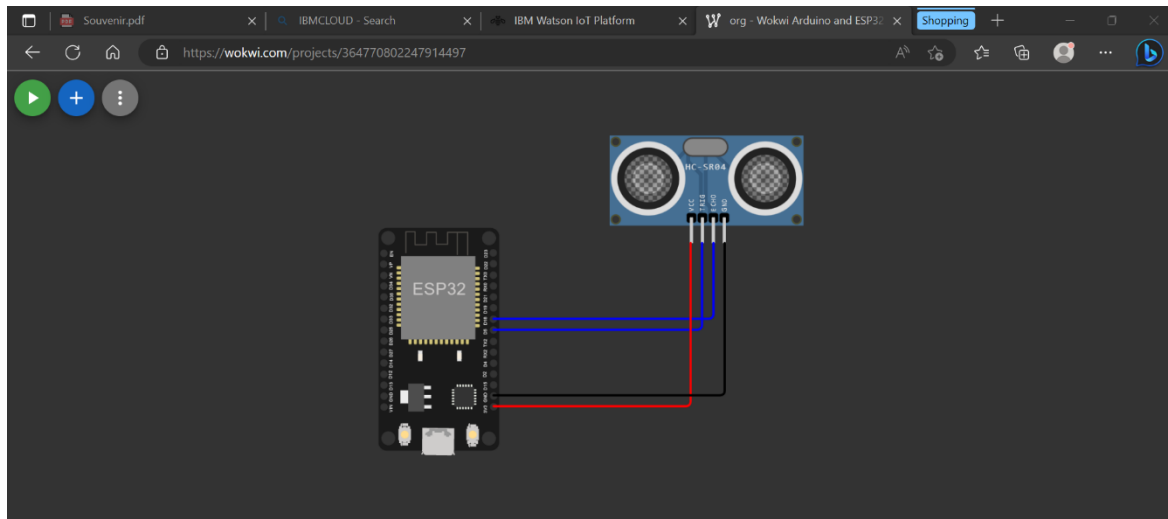
```

```

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++)
{
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}
}

```

Schematics :



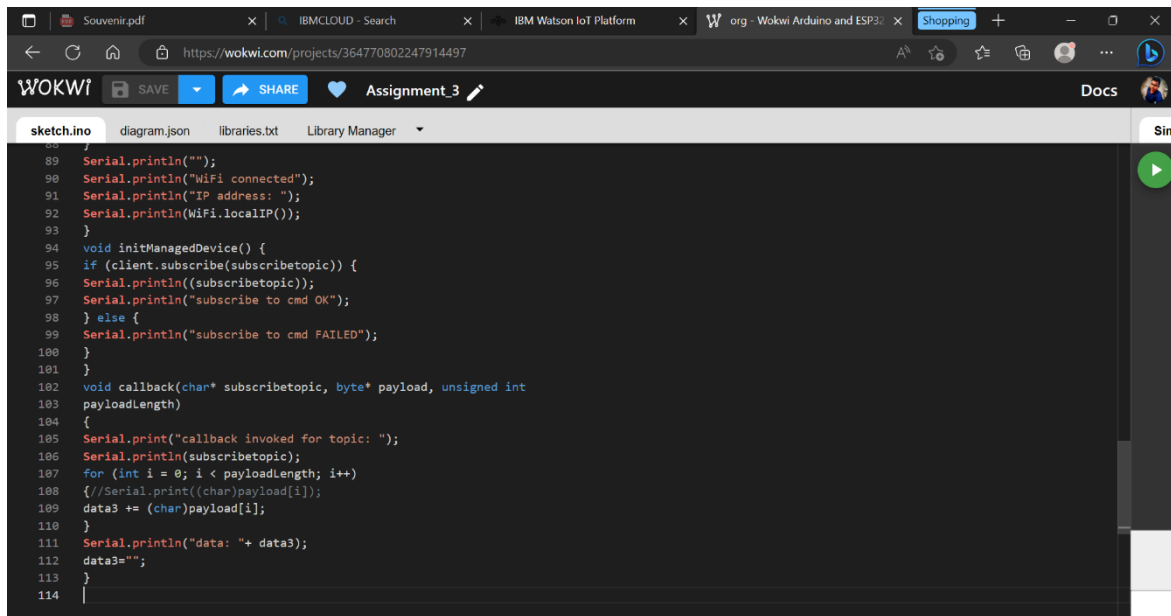
Screenshot of a Wokwi web IDE interface showing an Arduino sketch for an IoT project. The browser tabs include Souvenir.pdf, IBMCLCLOUD - Search, IBM Watson IoT Platform, Wokwi Arduino and ESP32, and Shopping. The URL is https://wokwi.com/projects/364770802247914497. The project is titled "Assignment_3".

The sketch code is as follows:

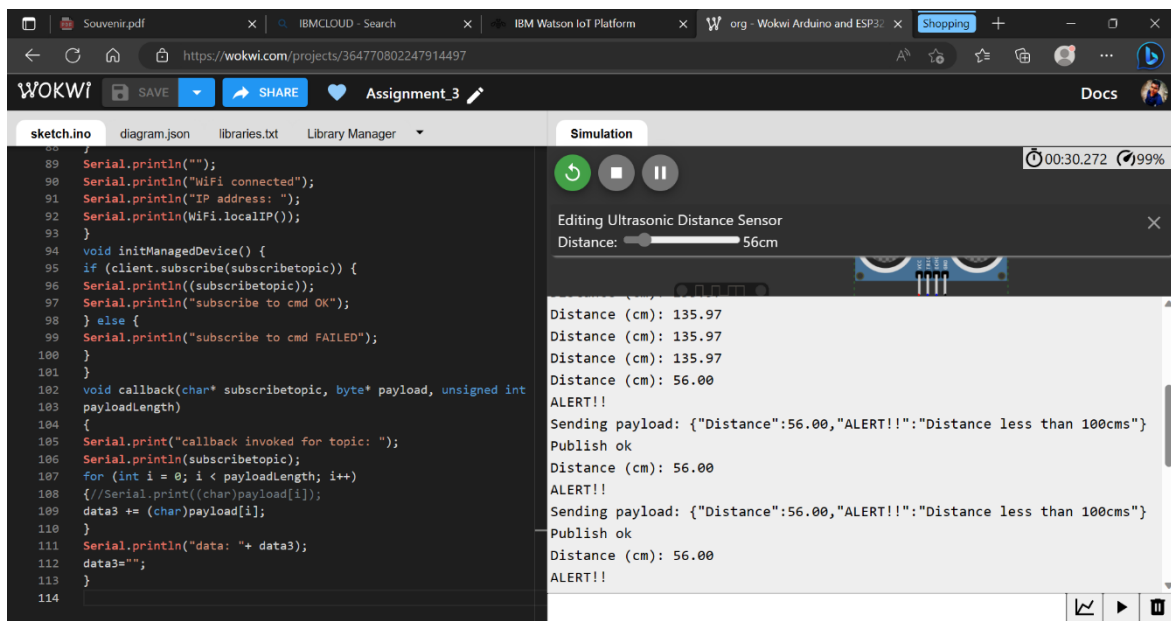
```
27 Serial.begin(115200);
28 pinMode(trigPin, OUTPUT);
29 pinMode(echoPin, INPUT);
30 wificonnect();
31 mqttconnect();
32 }
33 void loop()
34 {
35   digitalWrite(trigPin, LOW);
36   delayMicroseconds(2);
37   digitalWrite(trigPin, HIGH);
38   delayMicroseconds(10);
39   digitalWrite(trigPin, LOW);
40   duration = pulseIn(echoPin, HIGH);
41   distance = duration * SOUND_SPEED/2;
42   Serial.print("Distance (cm): ");
43   Serial.println(distance);
44   if(distance<100)
45   {
46     Serial.println("ALERT!!");
47     delay(1000);PublishData(distance);
48     delay(1000);
49     if (!client.loop()) {
50       mqttconnect();
51     }
52   }
53   delay(1000);}
54 }
```

The second screenshot shows the continuation of the sketch code:

```
55 void PublishData(float dist) {
56   mqttconnect();
57   String payload = "{\"Distance\": ";
58   payload += dist;
59   payload += ", \"ALERT!!\": \"\"Distance less than 100cms\"";
60   payload += "\"}";
61   Serial.print("Sending payload: ");
62   Serial.println(payload);
63   if (client.publish(publishTopic, (char*) payload.c_str())) {
64     Serial.println("Publish ok");
65   } else {
66     Serial.println("Publish failed");
67   }
68 }
69 void mqttconnect() {
70   if (!client.connected()) {
71     Serial.print("Reconnecting client to ");
72     Serial.println(server);
73     while (!client.connect(clientId, authMethod, token)) {
74       Serial.print(".");
75       delay(500);
76     }initManagedDevice();
77     Serial.println();
78   }
79 }
80 void wificonnect()
81 {
82   Serial.println();
83   Serial.print("Connecting to ");
84   WiFi.begin("Wokwi-GUEST", "", 6);
85   while (WiFi.status() != WL_CONNECTED) {
86     delay(500);
87     Serial.print(".");
88   }
89   Serial.println("");
90   Serial.println("WiFi connected");
91   Serial.println("IP address: ");
92   Serial.println(WiFi.localIP());
93 }
94 void initManagedDevice() {
95   if (client.subscribe(subscribetopic)) {
96     Serial.println((subscribetopic));
97     Serial.println("subscribe to cmd OK");
98   } else {
99     Serial.println("subscribe to cmd FAILED");
100   }
101 }
102 void callback(char* subscribetopic, byte* payload, unsigned int
103 payloadLength)
104 {
105   Serial.print("callback invoked for topic: ");
106   Serial.println(subscribetopic);
107   for (int i = 0; i < payloadLength; i++)
108     {Serial.print((char)payload[i]);
109     delay(10);
110   }
```



```
89 Serial.println("");
90 Serial.println("WiFi connected");
91 Serial.println("IP address: ");
92 Serial.println(WiFi.localIP());
93 }
94 void initManagedDevice() {
95   if (client.subscribe(subscribetopic)) {
96     Serial.println(subscribetopic);
97     Serial.println("subscribe to cmd OK");
98   } else {
99     Serial.println("subscribe to cmd FAILED");
100   }
101 }
102 void callback(char* subscribetopic, byte* payload, unsigned int
103 payloadLength)
104 {
105   Serial.print("callback invoked for topic: ");
106   Serial.println(subscribetopic);
107   for (int i = 0; i < payloadLength; i++)
108     //Serial.print((char)payload[i]);
109   data3 += (char)payload[i];
110 }
111 Serial.println("data: "+ data3);
112 data3="";
113 }
114 }
```

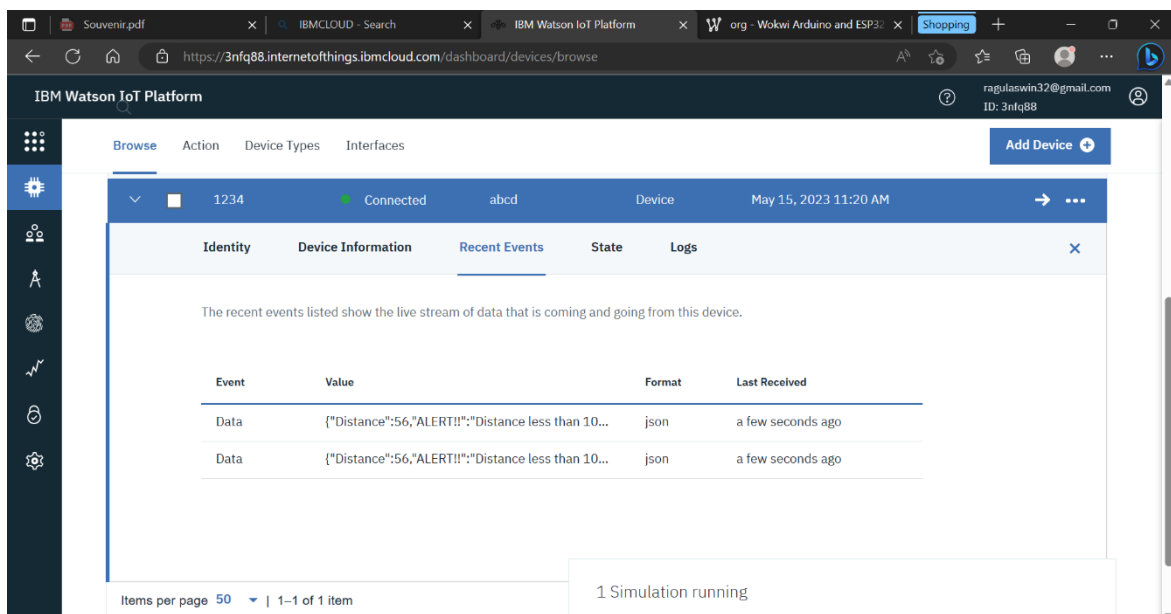


Simulation

00:30.272 99%

Editing Ultrasonic Distance Sensor
Distance: 56cm

Distance (cm): 135.97
Distance (cm): 135.97
Distance (cm): 135.97
Distance (cm): 56.00
ALERT!!
Sending payload: {"Distance":56.00,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 56.00
ALERT!!
Sending payload: {"Distance":56.00,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 56.00
ALERT!!



IBM Watson IoT Platform

3nfg88.internetofthings.ibmcloud.com/dashboard/devices/browse

ragulasewin32@gmail.com
ID: 3nfg88

1 Simulation running

Event	Value	Format	Last Received
Data	{"Distance":56,"ALERT!!":"Distance less than 10...	json	a few seconds ago
Data	{"Distance":56,"ALERT!!":"Distance less than 10...	json	a few seconds ago