

ASSIGNMENT 4

TEAM ID	PNT2022TMID36685
PROJECT NAME	Gas Leakage monitoring & Alerting system for Industries
TEAM MEMBER NAME	Baluprakash TS

Question:

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.

Solution :

Code:

```
// defines pins numbers
const int trigPin = 2;
const int echoPin = 5;
// defines variables
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in
  microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance= duration*0.034/2;
  // Prints the distance on the Serial Monitor
```

```

Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
if(distance <= 100){
Serial.println("Alert Distance is less than 100 cm");
}
}
}
DIAGRAM.JSON
{
"version": 1,
"author": "Uri Shaked",
"editor": "wokwi",
"parts": [
{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -21.91,
"left": -66.98, "attrs": {} },
{ "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -56.74,
"left": 85.25, "attrs": {} }
],
"connections": [
[ "esp:TX0", "$serialMonitor:RX", "", [] ],
[ "esp:RX0", "$serialMonitor:TX", "", [] ],
[ "ultrasonic1:ECHO", "esp:D5", "green", [ "v0" ] ],
[ "ultrasonic1:VCC", "esp:3V3", "red", [ "v99.32", "h-11.05" ] ],
[ "esp:GND.1", "ultrasonic1:GND", "black", [ "h0" ] ],
[ "esp:D2", "ultrasonic1:TRIG", "green", [ "h0" ] ]
]
}

```

OUTPUT:

IBM CLOUD OUTPUT:

The screenshot displays the IBM Watson IoT Platform interface. At the top, the header shows 'IBM Watson IoT Platform' and user information: '112819106001@smartinternz.com' and 'ID: bpy6g6'. Below the header, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a device card for '2468' with status 'Disconnected', 'Rey' as the device name, and 'Nov 3, 2022 6:24 PM' as the last update. Below the device card, there's a tabbed interface with 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events. The table has columns: 'Event', 'Value', 'Format', and 'Last Received'. It lists five events, all with the value '{"Distance":79,"Alert":"Distance less then 100"}' and format 'json'. At the bottom right, a status bar indicates '2 Simulations running'.

Event	Value	Format	Last Received
event_1	{"Distance":79,"Alert":"Distance less then 100"}	json	a few seconds ago
event_1	{"Distance":17,"Alert":"Distance less then 100"}	json	a few seconds ago
event_1	{"Distance":37,"Alert":"Distance less then 100"}	json	a few seconds ago
event_1	{"Distance":85,"Alert":"Distance less then 100"}	json	a few seconds ago
event_1	{"Distance":3,"Alert":"Distance less then 100"}	json	a few seconds ago

WOKWI
SAVE
SHARE
esp32-dht22.ino by urish
Docs SIGN IN

esp32-dht22.ino
diagram.json
libraries.txt
Library Manager

```

1  const int trigPin = 2;
2  const int echoPin = 5;
3  // defines variables
4  long duration;
5  int distance;
6  void setup() {
7    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
8    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
9    Serial.begin(9600); // Starts the serial communication
10 }
11 void loop() {
12   // Clears the trigPin
13   digitalWrite(trigPin, LOW);
14   delayMicroseconds(2);
15   // Sets the trigPin on HIGH state for 10 micro seconds
16   digitalWrite(trigPin, HIGH);
17   delayMicroseconds(10);
18   digitalWrite(trigPin, LOW);
19   // Reads the echoPin, returns the sound wave travel time in microseconds
20   duration = pulseIn(echoPin, HIGH);
21   // Calculating the distance
22   distance= duration*0.034/2;
23   // Prints the distance on the Serial Monitor
24   Serial.print("Distance: ");
25   Serial.print(distance);
26   Serial.println(" cm");
27   if(distance <= 100){
28     Serial.println("Alert Distance is less than 100 cm");
29   }
30 }
31 
```

Simulation
00:10.531 46%

Editing Ultrasonic Distance Sensor
Distance: 23cm

Distance: 23 cm
Alert Distance is less than 100 cm
Distance: 22 cm
Alert Distance is less than 100 cm
Distance: 23 cm
Alert Distance is less than 100 cm
Distance: 22 cm

WOKWI
SAVE
SHARE
esp32-dht22.ino by urish
Docs SIGN IN

esp32-dht22.ino
diagram.json
libraries.txt
Library Manager

```

1  {
2    "version": 1,
3    "author": "Uri Shaked",
4    "editor": "wokwi",
5    "parts": [
6      { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -53.33, "left": -99.5, "width": 100, "height": 100 },
7      { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -66.84, "left": 77.5, "width": 50, "height": 50 },
8    ],
9    "connections": [
10     [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
11     [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
12     [ "ultrasonic1:VCC", "esp:3V3", "red", [ "v0" ] ],
13     [ "ultrasonic1:TRIG", "esp:D2", "green", [ "v0" ] ],
14     [ "ultrasonic1:ECHO", "esp:D5", "green", [ "v0" ] ],
15     [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ]
16   ]
17 }

```

Simulation
00:12.831 42%

Editing Ultrasonic Distance Sensor
Distance: 59cm

Alert Distance is less than 100 cm
Distance: 58 cm
Alert Distance is less than 100 cm
Distance: 58 cm
Alert Distance is less than 100 cm
Distance: 59 cm
Alert Distance is less than 100 cm