Assignment -4
Wokwi Connection for the Ultrasonic Sensor

Assignment Date	07 November 2022
Team ID	PNT2022TMID44156
Student Name	Nabesha Fathima A
Student Roll Number	724019104016
Maximum Marks	2 Marks

Question:

Write code and connections in wokwi for the ultrasonic sensor.

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

Solution:

Program:

#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "4yi0vc"

#define DEVICE_TYPE "nodeMcu"

#define DEVICE_ID "Assignment4"

#define TOKEN "123456789"

#define speed 0.034

#define led 14

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

char publishTopic[] = "iot-2/evt/Data/fmt/json";

char topic[] = "iot-2/cmd/home/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=5;

```cpp
const int echopin=18;

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());
```

```cpp
}
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 = "{\"Normal Distance\":";
   payload += dist;
```

```cpp
    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<111){

   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("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="";

}
```
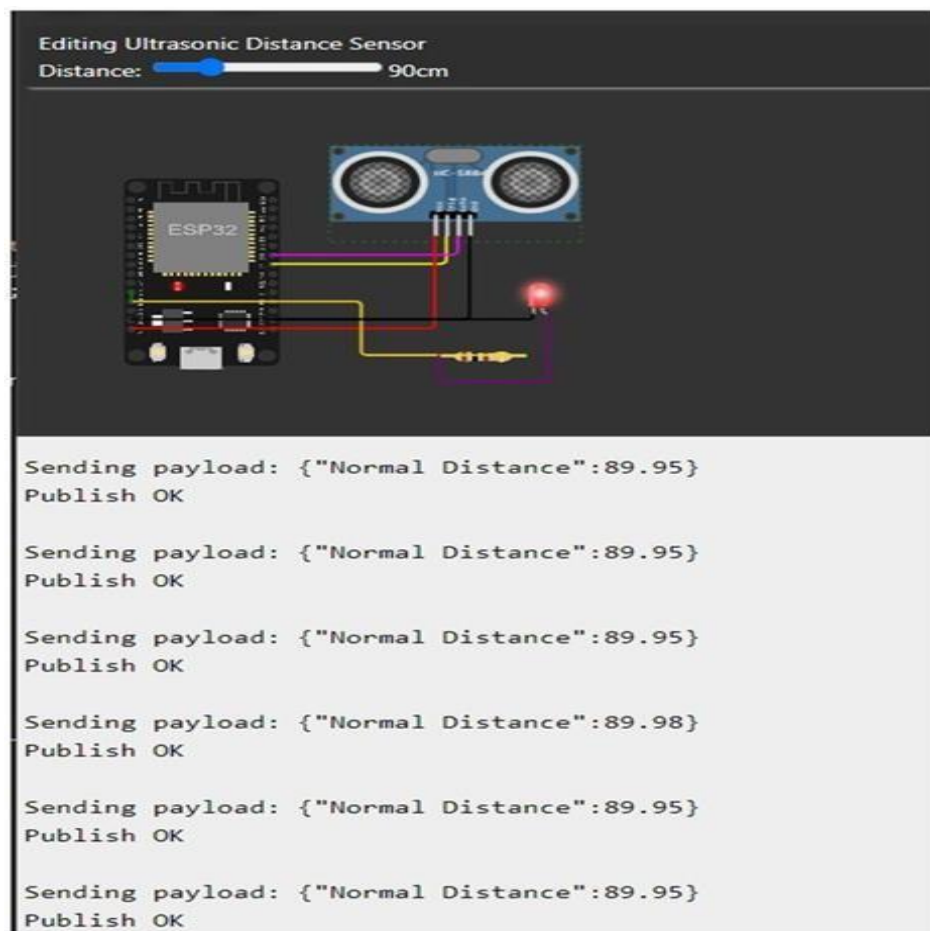
Output:

Case 1:

WHEN DISTANCE UNDER 100CM IT WILL SHOW NORMAL DISTANCE

Case 2:

WHEN DISTANCE CROSS 100CM IT WILL SHOW ALERT WITH WARNING MESSAGE DISTANCE

Editing Ultrasonic Distance Sensor
Distance: 107cm

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Case 3:

WHEN IT CROSS ABOVE 110CM IT TOTALLY MOVE TO IFF STATE ONCE IT REDUCE TO 110CM IT ON AGAIN

Editing Ultrasonic Distance Sensor
Distance: 125cm

Sending payload: {"Alert distance":106.96}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert distance":106.98}
Warning crosses 110cm -- it automaticaly of the loop

IBM Cloud Output

NORMAL DISTANCE

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	["Normal Distance":89.95]	json	a few seconds ago
Data	["Normal Distance":89.95]	json	a few seconds ago
Data	["Normal Distance":89.95]	json	a few seconds ago
Data	["Normal Distance":89.95]	json	a few seconds ago
Data	["Normal Distance":89.95]	json	a few seconds ago

ALERT DISTANCE

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	["Alert distance":106.98]	json	a few seconds ago
Data	["Alert distance":107.03]	json	a few seconds ago
Data	["Alert distance":106.98]	json	a few seconds ago
Data	["Alert distance":106.98]	json	a few seconds ago
Data	["Alert distance":106.98]	json	a few seconds ago