Problem statement :

IoT based safety gadget for child safety monitoring and notification.

Domain :

Internet of Things Assignment 4: Distance detection using ultrasonic sensor

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.

WOWKI LINK:

https://wokwi.com/projects/347931062880436818

BY,

SAMEERA N (623519106030)
MYTHILI T (623519106020)
YOGESHWARI V (623519106045)
SOWMIYA S (623519106038)

Solution:

```
#include<WiFi.h>
#include<WiFiClient.h>#inclu
de<PubSubClient.h>constinttr
igPin = 5; constintechoPin =
18;
//define sound speed in cm/uS
#define SOUND_SPEED
0.034#define CM_TO_INCH
0.393701long duration;
floatdistanceCm;
floatdistanceInch;


void callback(char* subscribetopic, byte* payload,unsignedintpayloadLength);   //-------
credentials of IBM Accounts------

#define ORG "da2mm3"//IBM ORGANITION ID

#define DEVICE_TYPE "eso32"//Device type mentioned in ibmwatson IOT Platform
#define DEVICE_ID "yogeshwari"//Device ID mentioned in ibmwatson IOT Platform
#define TOKEN "623519106045"//Token
String data3;



//-------- Customise the above values --------char server[] = ORG
".messaging.internetofthings.ibmcloud.com";// Server Namecharpublish Topic[] = "iot-
2/evt/Data/fmt/json";// topic name and type of event perform and format in which data
to be send
charsubscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND
COMMAND IS TEST OF FORMAT STRINGcharauthMethod[] = "use-token-auth";// authentication
methodchar token[] = TOKEN; charclientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;//client id
WiFiClientwifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient);

voidsetup() {
Serial.begin(115200); // Starts the serial
communicationpinMode(trigPin, OUTPUT); // Sets the trigPin
as an OutputpinMode(echoPin, INPUT); // Sets the echoPin as
an InputSerial.println();   wificonnect();   mqttconnect();

}
voidloop() {   // Clears the
trigPindigitalWrite(trigPin,
LOW);   delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro
secondsdigitalWrite(trigPin, HIGH);
delayMicroseconds(10);   digitalWrite(trigPin, LOW);
```

```
// Reads the echoPin, returns the sound wave travel time in microseconds   duration
= pulseIn(echoPin, HIGH);

// Calculate the distance
distanceCm = duration * SOUND_SPEED/2;

// Convert to inchesdistanceInch =
distanceCm * CM_TO_INCH;

// Prints the distance in the Serial Monitor
Serial.print("Distance (cm): ");
Serial.println(distanceCm);
Serial.print("Distance (inch): ");
Serial.println(distanceInch);

PublishData(distanceCm);
delay(1000);   if
(!client.loop()) {
mqttconnect();
  }
}    voidPublishData(float Cm) {
mqttconnect();//function call for connecting to ibm
/*     creating the String in in form JSon to update the data to ibm
cloud
  */
String payload = "{\"Distance (cm)\":";
payload += Cm;   payload += "}";


Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it
will print publish ok in Serial monitor or else it will print publish failed   } else
{
Serial.println("Publish failed");
  }
   } voidmqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");     Serial.println(server);
while(!!!client.connect(clientId, authMethod, token)) {
Serial.print(".");       delay(500);
    }
initManagedDevice();
Serial.println();
  } } voidwificonnect() //function defination for
wificonnect {
Serial.println();
Serial.print("Connecting to ");
```

```
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the
connectionwhile (WiFi.status() != WL_CONNECTED) {     delay(500);
Serial.print(".");
  }
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
voidinitManagedDevice() {
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,
unsignedintpayloadLength)  {

Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (inti = 0; i<payloadLength; i++) {
//Serial.print((char)payload[i]);     data3
+= (char)payload[i];
  }
}
```

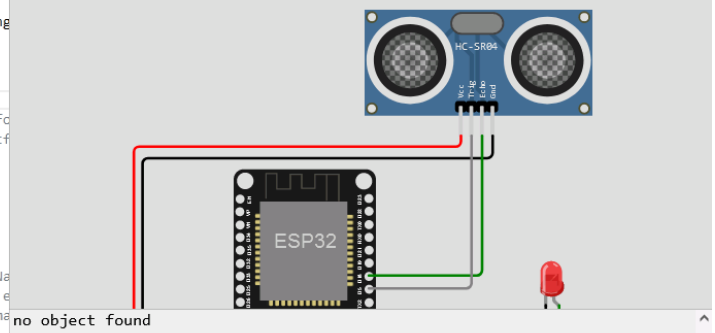
Output:

WOKWI SAVE SHARE sketch.ino Docs SIGN IN

sketch.ino ● diagram.json libraries.txt Library Manager

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt


void callback(char* subscribetopic, byte* payload, unsigned int payloadLeng

//-------credentials of IBM Accounts------

#define ORG "da2mm3"//IBM ORGANITION ID
#define DEVICE_TYPE "eso32"//Device type mentioned in ibm watson IOT Platfo
#define DEVICE_ID "yogeshwari"//Device ID mentioned in ibm watson IOT Platf
#define TOKEN "9994231139"      //Token
String data3;
float dist;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Na
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of e
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT comma
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the pred

int LED = 4;
```

Simulation 00:13.583 35%




```
no object found
Sending payload: {"distance":403.51,"object":"No"}
Publish ok
Distancein cm403.49
no object found
Sending payload: {"distance":403.49,"object":"No"}
Publish ok
```

IBM Watson IoT Platform 623519106045@smartinternz.com ID: da2mm3

Browse Action Device Types Interfaces Add Device

Device ID	Status	Device Type	Class ID	Date Added
yogeshwari	Connected	eso32	Device	10 Nov 2022 14:26

Identity Device Information Recent Events State Logs

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

Event	Value	Format	Last Received
Data	{"distance":403.49,"object":"No"}	json	a few seconds ago
Data	{"distance":403.49,"object":"No"}	json	a few seconds ago
Data	{"distance":403.49,"object":"No"}	json	a few seconds ago
Data	{"distance":403.49,"object":"No"}	json	a few seconds ago
Data	{"distance":403.45,"object":"No"}	json	a few seconds ago