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/347922168846221908

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 "tfltte"//IBM ORGANITION ID
#define DEVICE_TYPE "esp32"//Device type mentioned in ibmwatson IOT Platform
#define DEVICE_ID "mythilli"//Device ID mentioned in ibmwatson IOT Platform
#define TOKEN "8754774055 "//Token
String data3;



//-------- Customise the above values --------char server[] = ORG
".messaging.internetofthings.ibmcloud.com";// Server Namecharpublishtopic[] = "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];
  }
```