PNT2022TMID07016

Name: VISHNUKUMAR S Reg. No: GCTC1918148

Assignment – 4

ESP32 with Ultrasonic Senor and IoT Watson

# Task – 1:

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. Upload document with wokwi share link and images of IBM cloud

# Solution:

## Program:

#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

#define TRIG\_PIN 2 // ESP32 pin GIOP23 connected to Ultrasonic Sensor's TRIG pin

#define ECHO\_PIN 15 // ESP32 pin GIOP22 connected to Ultrasonic Sensor's ECHO pin

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

#define ORG "r3m467"//IBM ORGANITION ID

#define DEVICE\_TYPE "NalaiyaThiran"//Device type mentioned in ibm watson IOT Platform

#define DEVICE\_ID "NalaiyaThiran"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "NalaiyaThiran"     //Token

String data3;

float duration\_us, distance\_cm;

//-------- Customise the above values --------

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

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

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 ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32

{

**Serial**.begin(115200);

  pinMode(TRIG\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

  delay(10);

**Serial**.println();

  wificonnect();

  mqttconnect();

}

void loop()// Recursive Function

{

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG\_PIN, LOW);

  // measure duration of pulse from ECHO pin

  duration\_us = pulseIn(ECHO\_PIN, HIGH);

  // calculate the distance

  distance\_cm = 0.017 \* duration\_us;

  // print the value to Serial Monitor

**Serial**.print("distance: ");

**Serial**.print(distance\_cm);

**Serial**.println(" cm");

  delay(500);

  if(distance\_cm<100){

    PublishData(distance\_cm);

  }

  delay(1000);

  if (!client.loop()) {

    mqttconnect();

  }

}

/\*.....................................retrieving to Cloud...............................\*/

void PublishData(float distance\_cm) {

  mqttconnect();//function call for connecting to ibm

  /\*

     creating the String in in form JSon to update the data to ibm cloud

  \*/

  String payload = "{\"Alert\":\"ON\"";

  payload += "," "\"Distance\_cm\":";

  payload += distance\_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");

  }

}

void mqttconnect() {

  if (!client.connected()) {

**Serial**.print("Reconnecting client to ");

**Serial**.println(server);

    while (!!!client.connect(clientId, authMethod, token)) {

**Serial**.print(client.connect(clientId, authMethod, token));

**Serial**.print(".");

      delay(500);

    }

**Serial**.println();

  }

}

void wificonnect() //function defination for wificonnect

{

**Serial**.println();

**Serial**.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.println("WiFi connected");

**Serial**.println("IP address: ");

**Serial**.println(WiFi.localIP());

}

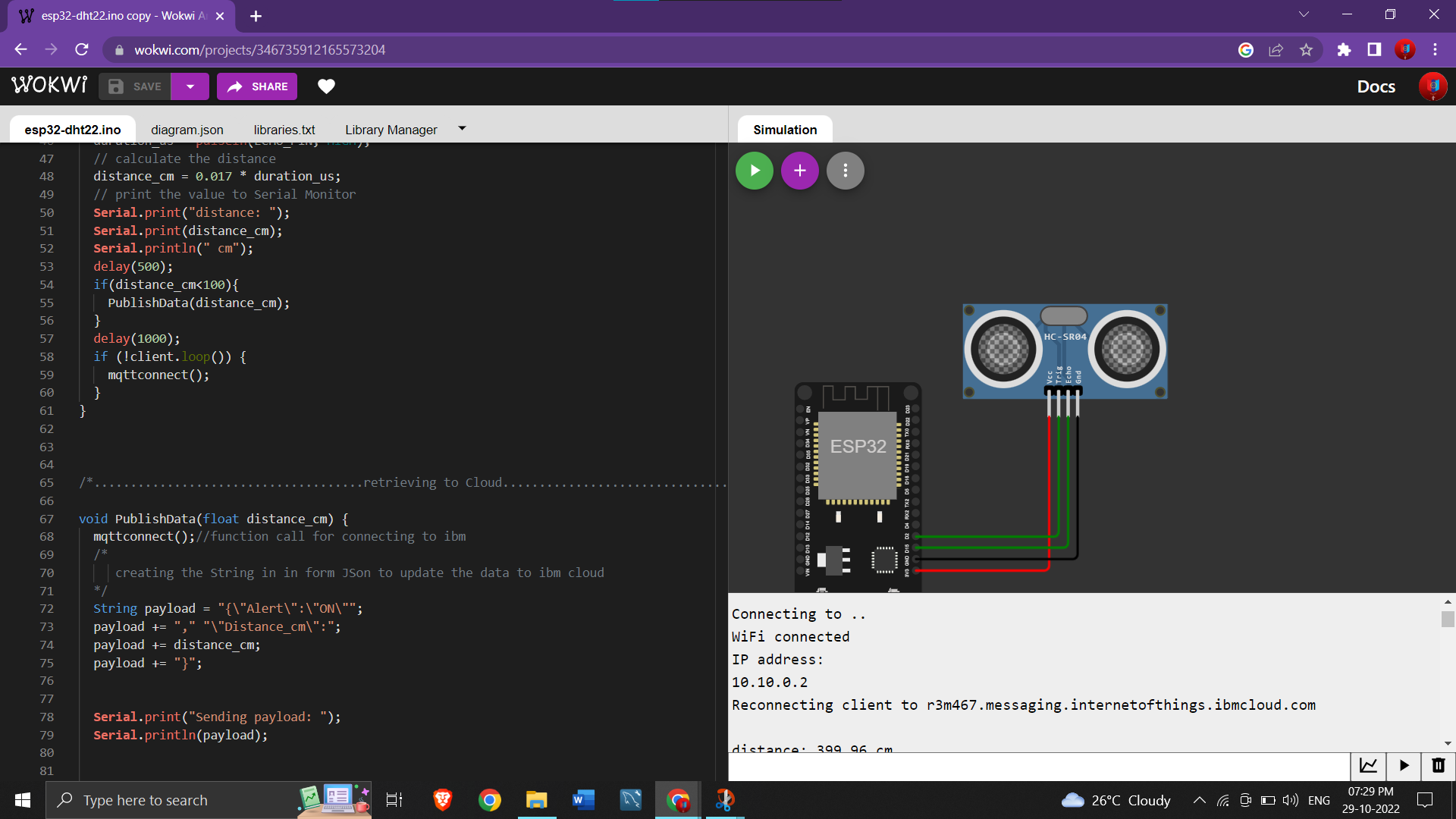
## Output of Program:

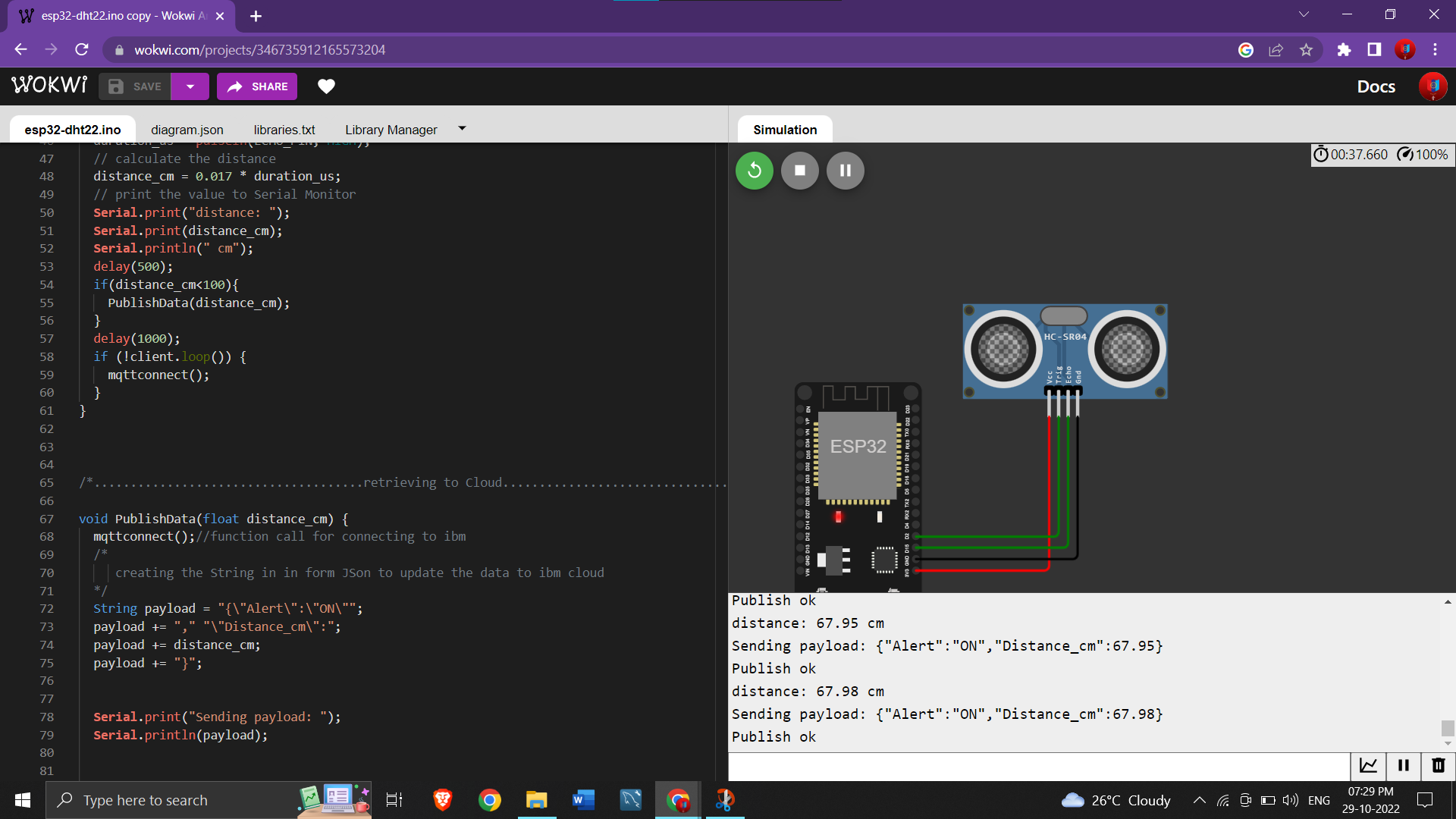
Wokwi Project Link:

<https://wokwi.com/projects/346735912165573204>

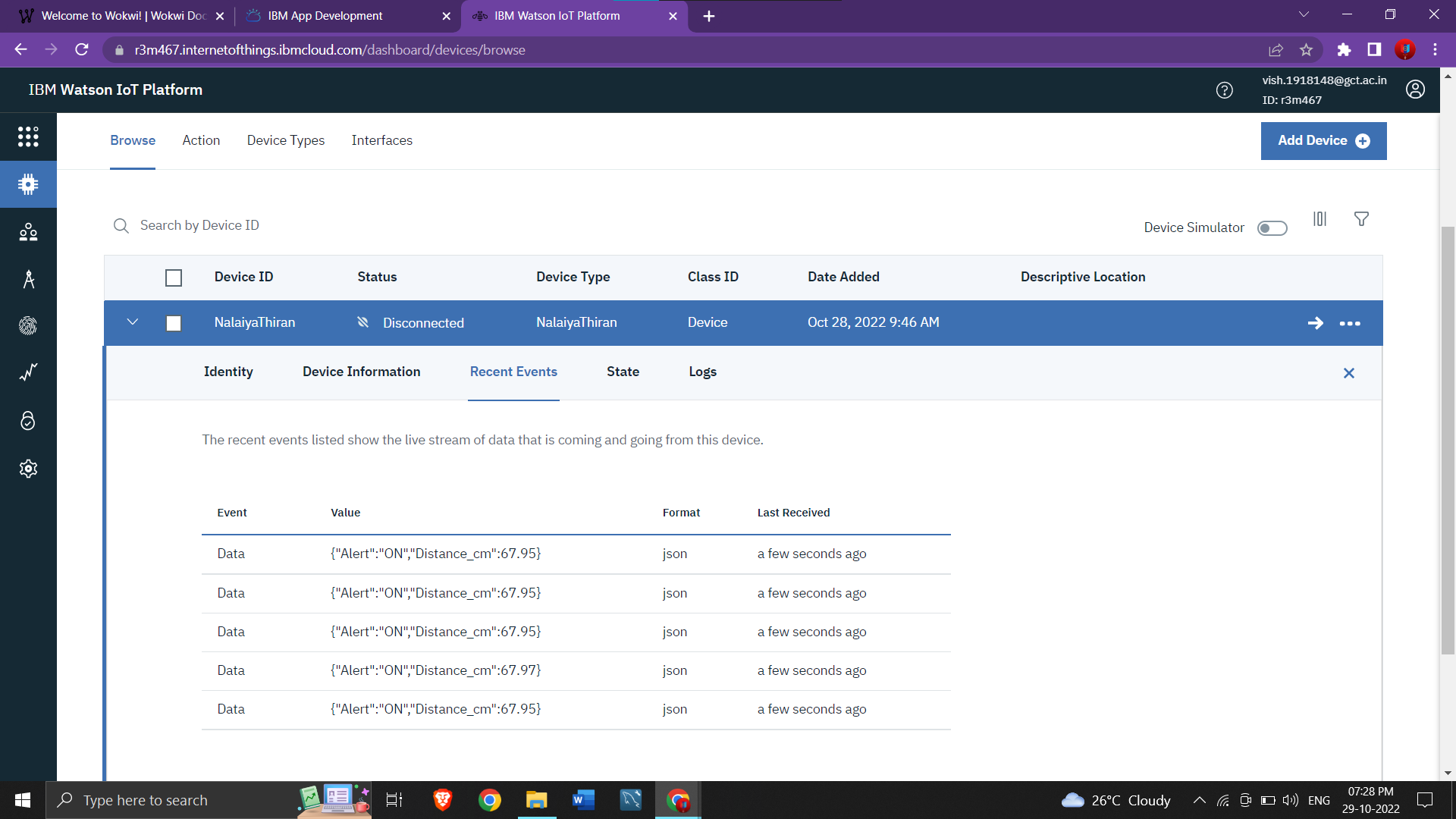
Note: The IBM Cloud Credentials of My Device is replaced with \* please include your IBM Cloud Device credentials before running the Simulation

## Wokwi Website Screenshot:





IBM Cloud Screenshot:



## Explanation of Program:

Initially, we have imported the <Wifi.h> and <PubSubClient.h> header files as they are needed to connect wifi and MQTT Protocol. Then, Define Trigger pin and Echo pin values where the ultrasonic sensor is connected with ESP32 module. Then, Define the IBM Account Credentials such as ORG, Device\_Type, Device\_ID and Token. Also define the server, publishTopic, SubscribeTopic, authMethod, Token and ClientID. Create Object for WifiClient and PubSubClient.

Then, Start the void Setup() Function, Begin the Serial Monitor and set PinMode of Trigger Pin as Output and Echo Pin as Input and call wificonnect() and mqttconnect() to initialize wifi and mqtt Connection and Define their methods to make the Connection.

Then, Begin the void loop() function, digitalWrite HIGH to Trigger Pin and create a delay of 10 microseconds and write back LOW. Then, use pulseIn() function with Echo Pin to calculate the Duration and calculate the distance. If the Distance is less than 100cm call PublishData() Function to publish the Data to IoT Watson Device.

Finally, Define the PublishData() function with message as parameter. Then Define string that contains the payload with the message to be sent in the Json Format. Call Client.publish() function with publishTopic and payload as parameter. Also define the wificonnect() and mqttconnect() to make intial connection with wifi and mqtt connection.