# Assignment-4

**Question-1:**

Writecodeandconnectionsinwokwiforultrasonicsensor.Wheneverdistanceislessthan100cms

send“alert”toibmcloudanddisplayindevicerecent events.

Uploaddocumentwithwokwisharelinkandimagesofibmcloud.

# Solution:

#include<WiFi.h>//libraryforwifi#include<PubSubClient.h>//libraryforMQtt

#defineECHO\_GPIO12

#defineTRIGGER\_GPIO 13

#define MAX\_DISTANCE\_CM 100 // Maximum of 5 meters#include"Ultrasonic.h"

Ultrasonic ultrasonic(13, 12); intdistance;

voidcallback(char\*subscribetopic,byte\*payload,unsignedintpayloadLength);

//-------credentialsofIBMAccounts------

#defineORG"8zenx2"//IBM ORGANITIONID

#define DEVICE\_TYPE "raspberrypi"//Device type mentioned in ibm watson IOTPlatform #defineDEVICE\_ID "dhaya29"//Device ID mentioned in ibm watson IOTPlatform#defineTOKEN"Dhaya@29"

//TokenStringdata3;floath,t;

//-------- Customise the above values --------char server[] = ORG".messaging.internetofthings.ibmcloud.com";//ServerName

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform andformatinwhichdatatobesend

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type ANDCOMMAND IS TEST OF FORMAT STRINGchar authMethod[] = "use-token-auth";// authenticationmethodchartoken[]=TOKEN;

charclientId[] ="d:"ORG":"DEVICE\_TYPE":"DEVICE\_ID;//clientid

//-

WiFiClientwifiClient;//creatingtheinstanceforwificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id bypassingparameterlikeserverid,portandwificredential

voidsetup()//configureingtheESP32 {

**Serial**.begin(115200);delay(10);**Serial**.println();wificonnect();mqttconnect();

}

voidloop()//RecursiveFunction

{

distance=ultrasonic.read(CM); if(distance

<100){

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

**Serial**.println(distance)

; PublishData(distance);delay(1000);if(!client.loop()){

mqttconnect();

}

}

delay(1000);

}

/\*.....................................retrievingtoCloud \*/

voidPublishData(floattemp){mqttconnect();//functioncallforconnectingtoibm

/\*creatingtheStringininform JSontoupdatethe data to ibmcloud

\*/

Stringpayload ="{\"Alert Distance:\":";payload+=temp;payload+="}";

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

**Serial**.println(payload);

if(client.publish(publishTopic,(char\*)payload.c\_str())){

**Serial**.println("Publishok");// ifitsucessfullyupload dataonthecloudthen it willprintpublishokinSerialmonitororelseitwillprintpublishfailed

}else{

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

}

}

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();

} } void wificonnect() //function defination forwificonnect

{

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

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("subscribetocmdOK");

}else{

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

}

}

voidcallback(char\*subscribetopic, byte\*payload,unsignedintpayloadLength)

{

**Serial**.print("callback invoked for topic: ");**Serial**.println(subscribetopic); for (int i =0;i<payloadLength;i++){

//Serial.print((char)payload[i]);data3+=(char)payload[i];

}

**Serial**.println("data: "+data3);if(data3=="lighton"){

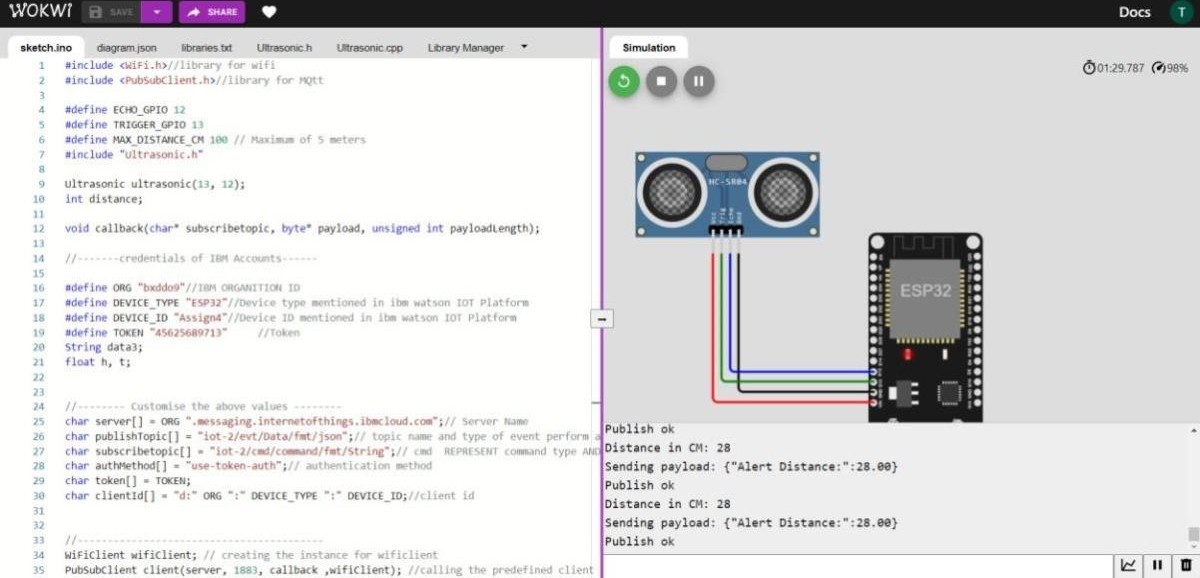
**Serial**.println(data3);

}else

{

**Serial**.println(data3);

}data3="";

}

