

NAME : THAMEEM S

REG NO : 513219104304

PROGRAM

Smart Waste Management System for Metropolitan Cities

ASSIGNMENT 4:

Write code and connections in wokwi for ultrasonic sensors. Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events. Uplode document with wokwi share link

and images of ibm cloud.

CODE:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "ztcz45"

#define DEVICE_TYPE "naveen"

#define DEVICE_ID "naveen123"

#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;

const int echopin=18;

String command;

String data="";

long duration;

float dist;

void setup()
{
    Serial.begin(115200);

    pinMode(led, OUTPUT);

    pinMode(trigpin, OUTPUT);

    ...

[10:32 pm, 23/10/2022] Gogul B.E CSE: }

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;

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\\":\n";

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:

The image shows a Wokwi IDE window on the left and a TTP IoT Platform web interface on the right.

Wokwi IDE: The code editor shows a C++ sketch for an Arduino Uno. The sketch defines a distance sensor (HC-SR04) and a loop that sends distance data to a cloud server. The simulation window shows a breadboard with the sensor and an Arduino Uno. The console output shows the following messages:

```

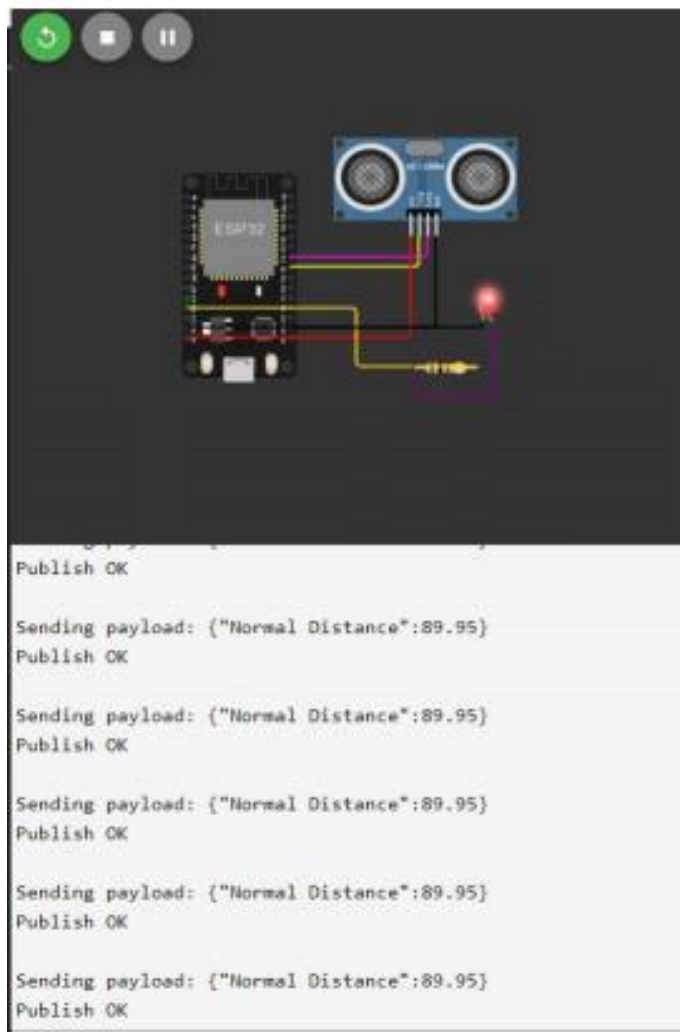
Sending payload: {"Alert distance":118.96}
Warning crosses 118cm -- It automatically of the loop
Sending payload: {"Alert distance":118.96}
Warning crosses 118cm -- It automatically of the loop
Sending payload: {"Alert distance":118.96}
Warning crosses 118cm -- It automatically of the loop
Sending payload: {"Alert distance":118.96}
Warning crosses 118cm -- It automatically of the loop

```

TTP IoT Platform: The web interface shows a list of devices. The device "esp8266" is shown with a status of "Connected". The "Recent Events" tab is selected, showing a table of data points:

Event	Value	Format	Last Received
Data	["Alert distance":118.96]	json	a few seconds
Data	["Alert distance":118.96]	json	a few seconds
Data	["Alert distance":118.96]	json	a few seconds
Data	["Alert distance":118.96]	json	a few seconds
Data	["Alert distance":118.96]	json	a few seconds

1. When distance under 100 cm it wil show normal distance.



2. When distance cross 100 cm it will show ALERT warning message distance

The screenshot displays the Wokwi IDE on the left, showing a C++ program for an Arduino Uno. The program uses an ultrasonic sensor to measure distance and sends data to the IBM Watson IoT Platform. The code includes comments and function calls for setting up the sensor, connecting to the IoT platform, and sending data. The right side of the image shows the IBM Watson IoT Platform interface, specifically the 'Recent Events' tab for a device named 'THAMEEN123'. The table below shows the recent events received from the device.

Event	Value	Format	Last Received
Data	{"Alert distance":110.90}	json	a few second
Data	{"Alert distance":110.90}	json	a few second
Data	{"Alert distance":110.90}	json	a few second
Data	{"Alert distance":110.90}	json	a few second
Data	{"Alert distance":110.90}	json	a few second

3. When it cross above 110 cm it today move to iff state once it

reduce to 110 it on again

Connection information:

Basic connection information about this device.

Organization ID : ztcz45

Device Type : THAMEEN

Device ID :THAMEEN123

Authentication Method : use-token-auth Authentication Token : 123456789

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	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	
Data	{"Normal Distance":89.95}	json	a few second	