

REAL TIME RIVER WATER QUALITY
MONITORING AND CONTROL SYSTEM

SUBMITTED BY

SOWMYA A (113219041114)

BACHELOR OF ENGINEERING
IN ELECTRONICS
AND COMMUNICATION ENGINEERING

ASSIGNMENT-04

Question:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cm send "alert" to IBM Cloud and display in device recent events.

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

Solution:

```
#include<WiFi.h>//libraryforwifi#include<PubSubClient.h>//libraryforMQTT

#defineECHO_GPIO12
#defineTRIGGER_GPIO13
#defineMAX_DISTANCE_CM100//Maximumof5meters#include"Ultrasonic.h"

Ultrasonicultrasonic(13,12);intdistance;

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

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

#defineORG"dv1snq"//IBMORGANIZATIONID
#defineDEVICE_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform#define
DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT
Platform#defineTOKEN"45682367915" //Token
Stringdata3;floatt;

//-----Customisetheabovevalues-----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
Namechar publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type
ofeventperform andformatinwhich datato besend
charsubscribetopic[]="iot-2/cmd/command/fmt/String";//cmdREPRESENT
commandtype ANDCOMMANDISTESTOFFORMAT STRING
```

```

char authMethod[] = "use-token-auth";// authentication
methodchar token[]=TOKEN;
char clientId[]="d:ORG":"DEVICE_TYPE ":"DEVICE_ID";//clientid

//-----
WiFiClient wifiClient; // creating the instance for
wificlientPubSubClient client(server, 1883, callback ,wifiClient);
//calling the predefined client id by passing parameter like server
id,port and wificredential

void setup()//configureing the ESP32
{
    Serial.begin(115200);
    delay(10);Serial.println();
    wifiConnect();
    mqttConnect();
}

void loop()//RecursiveFunction
{
    distance =
    ultrasonic.read(CM);if(distance
    <100){Serial.print("Distance in
    CM:
    ");Serial.println(distance);Publi
    shData(distance);delay(1000);
    if (!client.loop())
        {mqttConnect();
        }

    }

    delay(1000);

}

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

void PublishData(float temp)
{mqttConnect();//function call for connecting to ibm
/*
    creating the String in inform JS ontoupdate the data to ibmcloud
    */
    String payload="{\"AlertDistance\":\"";

```

```

payload +=
temp;payload+="}
";

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

if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("Publish ok");// if it successfully upload data on the
cloudthen it will print publish ok in Serial monitor or else it will print
publishfailed
}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();
}
}

voidwificonnect();//functiondefinationforwificonnect
{
    Serial.println();Serial.print("Co
nnectingto");

    WiFi.begin("Wokwi-
GUEST","",6);//passingthewificredentialstoestablishtheconnection
    while (WiFi.status() != WL_CONNECTED)
    {delay(500);
        Serial.print(".");
    }
    Serial.println("");Serial.println
("WiFi
connected");Serial.println("IP
address:
");Serial.println(WiFi.localIP())
;
}

```

```

void initManagedDevice(){
    if (client.subscribe(subscribetopic))
        {Serial.println((subscribetopic));Serial.println("subscribetocmdOK");}
    else{
        Serial.println("subscribetocmdFAILED");
    }
}

void callback(char*subscribetopic,byte*payload,unsigned intpayloadLength)
{
    Serial.print("callbackinvokedfortopic:");
    Serial.println(subscribetopic);
    for(int i=0;i<payloadLength;i++){
        //Serial.print((char)payload[i]);data
        3+= (char)payload[i];
    }
    Serial.println("data:"+data3);if
    (data3=="lighton")
    {
        Serial.println(data3);
    }
    else
    {
        Serial.println(data3);
    }
    data3="";
}

```

Wokwi

link:<https://wokwi.com/projects/346659959540286034>

WOKWI

SAVE

SHARE

Docs

sketch.ino

diagram.json

libraries.txt

Ultrasonic.h

Ultrasonic.cpp

Library Manager

Simulation

callback

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 #define ECHO_GPIO 12
5 #define TRIGGER_GPIO 13
6 #define MAX_DISTANCE_CM 100 // Maximum of 5 meters
7 #include "Ultrasonic.h"
8
9 Ultrasonic ultrasonic(13, 12);
10 int distance;
11
12 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
13
14 //-----credentials of IBM Accounts-----
15
16 #define ORG "dvisng" //IBM ORGANIZATION ID
17 #define DEVICE_TYPE "ESP32" //Device type mentioned in ibm watson IOT Platform
18 #define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
19 #define TOKEN "45682367915" //Token
20 String data3;
21 float h, t;
22
23 //----- Customise the above values -----
24
25 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
26 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform
27 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
28 char authMethod[] = "use-token-auth"; // authentication method
29 char token[] = TOKEN;
30 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
31

```

WOKWI

SAVE

SHARE

Docs

sketch.ino

diagram.json

libraries.txt

Ultrasonic.h

Ultrasonic.cpp

Library Manager

Simulation

callback

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 #define ECHO_GPIO 12
5 #define TRIGGER_GPIO 13
6 #define MAX_DISTANCE_CM 100 // Maximum of 5 meters
7 #include "Ultrasonic.h"
8
9 Ultrasonic ultrasonic(13, 12);
10 int distance;
11
12 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
13
14 //-----credentials of IBM Accounts-----
15
16 #define ORG "dvisng" //IBM ORGANIZATION ID
17 #define DEVICE_TYPE "ESP32" //Device type mentioned in ibm watson IOT Platform
18 #define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
19 #define TOKEN "45682367915" //Token
20 String data3;
21 float h, t;
22
23 //----- Customise the above values -----
24
25 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
26 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform
27 char subscribtopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
28 char authMethod[] = "use-token-auth"; // authentication method
29 char token[] = TOKEN;
30 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
31

```

IBM Watson IoT Platform

2019504510@smartinternz.com

ID: dv1snq

Browse

Action

Device Types

Interfaces

Add Device

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

| Event | Value | Format | Last Received |
|-------|-----------------------|--------|-------------------|
| Data | {"Alert Distance":62} | json | a few seconds ago |
| Data | {"Alert Distance":62} | json | a few seconds ago |
| Data | {"Alert Distance":62} | json | a few seconds ago |
| Data | {"Alert Distance":62} | json | a few seconds ago |
| Data | {"Alert Distance":62} | json | a few seconds ago |

Items per page 50

1-1 of 1 item

1 of 1 page

1

WOKWI

SAVE SHARE

Docs

sketch.ino diagram.json libraries.txt Ultrasonic.h Ultrasonic.cpp Library Manager

> callback Aa Alt 7 of 4

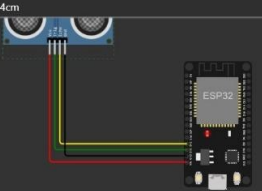
```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for mqtt
3
4 #define ECHO_GPIO 12
5 #define TRIGGER_GPIO 13
6 #define MAX_DISTANCE_CM 100 // Maximum of 5 meters
7 #include "Ultrasonic.h"
8
9 Ultrasonic ultrasonic(13, 12);
10 int distance;
11
12 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
13
14 //.....credentials of IBM Accounts.....
15
16 #define ORG "dv1snq" //IBM ORGANIZATION ID
17 #define DEVICE_TYPE "ESP32" //Device type mentioned in ibm watson IOT Platform
18 #define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
19 #define TOKEN "45682367915" //Token
20 String data3;
21 float h, t;
22
23
24 //..... Customise the above values .....
25 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
26 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform
27 char subscribetopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command type
28 char authMethod[] = "use-token-auth"; // authentication method
29 char token[] = TOKEN;
30 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
31
```

Simulation

00:39:161 100%

Editing Ultrasonic Distance Sensor

Distance: 124cm



Publish ok

Distance in CM: 56

Sending payload: {"Alert Distance":"56.00"}

Publish ok

Distance in CM: 56

Sending payload: {"Alert Distance":"56.00"}

Publish ok