ASSIGNMENT-4

**UltrasonicsensorsimulationinWokwi**

|  |  |  |  |
| --- | --- | --- | --- |
| TeamID |  | PNT2022TM11782 |  |
|  | | |
| Title | GasLeakageMonitoringandAlertingSystem | | |
| Name | SANJEEVI GG | | |
| MinimumMarks | 2Marks | | |

**Question:**

Write a code and connections in Wokwi for the Ultrasonic sensor.Wheneverthedistanceislessthan100cmssendan“Alert”toIBMcloudanddisplay inthedevicerecentevents.

# Code:

#include<WiFi.h>#include<PubSubClient.h>

void callback(char\* subscribetopic, byte\* payload, unsigned intpayloadLength);

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

#defineORG"d19wub"//IBMORGANITIONID

#defineDEVICE\_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform

#define DEVICE\_ID "3C-91-80-49-01-C9"//Device ID mentioned in ibm watson IOT Platform#defineTOKEN"cE&QcASnabqYe18-1f"//Token

Stringdata3;

char server[] = ORG ".[messaging.internetofthings.ibmcloud.com](http://messaging.internetofthings.ibmcloud.com/)";charpublishTopic[]="iot-2/evt/Data/fmt/json";

char subscribetopic[] = "iot-2/cmd/test/fmt/String";charauthMethod[]="use-token-auth";

chartoken[]=TOKEN;

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

WiFiClientwifiClient;

PubSubClient client(server, 1883, callback ,wifiClient);constinttrigPin=5;

const int echoPin = 18;#define SOUND\_SPEED 0.034longduration;

float distance;voidsetup(){

**Serial**.begin(115200);pinMode(trigPin,OUTPUT);pinMode(echoPin, INPUT);wificonnect();mqttconnect();

}

voidloop()

{

digitalWrite(trigPin, LOW);delayMicroseconds(2);digitalWrite(trigPin, HIGH);delayMicroseconds(10);digitalWrite(trigPin,LOW);duration = pulseIn(echoPin, HIGH);distance = duration \* SOUND\_SPEED/2;**Serial**.print("Distance (cm): ");**Serial**.println(distance);if(distance>100)

{

**Serial**.println("ALERT!!");delay(100);PublishData(distance);delay(100);

if (!client.loop()) {mqttconnect();

}

}

delay(100);

}

void PublishData(float dist) {mqttconnect();

String payload = "{\"Distance\":";payload+=dist;

payload += ",\"ALERT!!\":""\"Distance less than 100cms\"";payload+="}";

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

**Serial**.println(payload);

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

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

} else{

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

}

}

voidmqttconnect(){

if (!client.connected()) {**Serial**.print("Reconnectingclientto");**Serial**.println(server);

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

**Serial**.print(".");delay(100);

}

initManagedDevice();

**Serial**.println();

}

}

voidwificonnect()

{

**Serial**.println();**Serial**.print("Connecting to ");WiFi.begin("Wokwi-GUEST","",6);while (WiFi.status() != WL\_CONNECTED) {delay(100);

**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("callbackinvokedfortopic:");

**Serial**.println(subscribetopic);

for(inti=0;i<payloadLength;i++){

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

}

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

}

#include<WiFi.h>#include<PubSubClient.h>

void callback(char\* subscribetopic, byte\* payload, unsigned intpayloadLength);

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

#defineORG"d19wub"//IBMORGANITIONID

#defineDEVICE\_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform

#define DEVICE\_ID "3C-91-80-49-01-C9"//Device ID mentioned in ibm watson IOT Platform#defineTOKEN"cE&QcASnabqYe18-1f"//Token

Stringdata3;

char server[] = ORG ".[messaging.internetofthings.ibmcloud.com](http://messaging.internetofthings.ibmcloud.com/)";charpublishTopic[]="iot-2/evt/Data/fmt/json";

char subscribetopic[] = "iot-2/cmd/test/fmt/String";charauthMethod[]="use-token-auth";

chartoken[]=TOKEN;

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

WiFiClientwifiClient;

PubSubClient client(server, 1883, callback ,wifiClient);constinttrigPin=5;

const int echoPin = 18;#define SOUND\_SPEED 0.034longduration;

float distance;voidsetup(){

**Serial**.begin(115200);pinMode(trigPin,OUTPUT);pinMode(echoPin, INPUT);wificonnect();mqttconnect();

}

voidloop()

{

digitalWrite(trigPin, LOW);delayMicroseconds(2);digitalWrite(trigPin, HIGH);delayMicroseconds(10);digitalWrite(trigPin,LOW);duration = pulseIn(echoPin, HIGH);distance = duration \* SOUND\_SPEED/2;**Serial**.print("Distance (cm): ");**Serial**.println(distance);if(distance>100)

{

**Serial**.println("ALERT!!");delay(100);PublishData(distance);delay(100);

if (!client.loop()) {mqttconnect();

}

}

delay(100);

}

void PublishData(float dist) {mqttconnect();

String payload = "{\"Distance\":";payload+=dist;

payload += ",\"ALERT!!\":""\"Distance less than 100cms\"";payload+="}";

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

**Serial**.println(payload);

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

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

} else{

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

}

}

voidmqttconnect(){

if (!client.connected()) {**Serial**.print("Reconnectingclientto");**Serial**.println(server);

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

**Serial**.print(".");delay(100);

}

initManagedDevice();

**Serial**.println();

}

}

voidwificonnect()

{

**Serial**.println();**Serial**.print("Connecting to ");WiFi.begin("Wokwi-GUEST","",6);while (WiFi.status() != WL\_CONNECTED) {delay(100);

**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("callbackinvokedfortopic:");

**Serial**.println(subscribetopic);

for(inti=0;i<payloadLength;i++){

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

}

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

}

# Diagram.json:

{

"version":1,

"author": "sweetysharon","editor": "wokwi","parts":[

{"type":"wokwi-esp32-devkit-v1","id":"esp","top":-4.67,"left":-112.87,"attrs":

{}},

{"type":"wokwi-hc-sr04","id":"ultrasonic1","top":15.96,"left":89.17,"attrs":

{}}

],

"connections":[

["esp:TX0","$serialMonitor:RX","",[]],

[ "esp:RX0", "$serialMonitor:TX", "", [] ],[

"esp:VIN","ultrasonic1:VCC","red",

["h-37.16","v-178.79","h200","v173.33","h100.67"]

],

["esp:GND.1","ultrasonic1:GND","black",["h39.87","v44.04","h170"]],

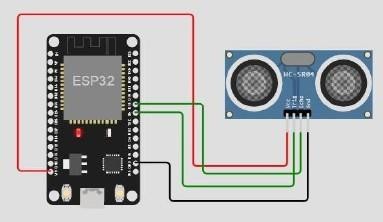
["esp:D5","ultrasonic1:TRIG","green",["h54.54","v85.07","h130.67"]],

["esp:D18","ultrasonic1:ECHO","green",["h77.87","v80.01","h110"]]

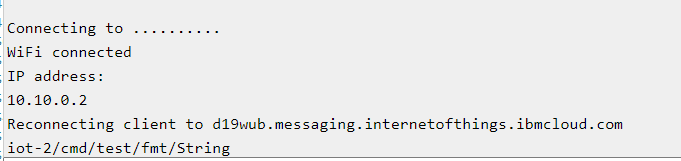
]

}

# CircuitDiagram:



**Output:**



# IBMCloudOutput:

