***Assignment-4***

# Write code and connections in wokwi for ultrasonic sensor.

# Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

***Solution code:***

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

#include <PubSubClient.h>//library for MQtt #define ORG "q1wscz”

#define DEVICE\_E "sampledevice" #define DEVICE\_D "24052002

#define TOKEN "K9)lI1C@tX6yO(J6L1" const int T\_PIN = 5;

const int E\_PIN = 4;

//-------- 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\_E ":" DEVICE\_D;//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() {

Serial.begin(115200); pinMode(T\_PIN, OUTPUT); pinMode(E\_PIN, INPUT); wificonnect(); mqttconnect();

}

float readDistanceCM() { digitalWrite(T\_PIN, LOW); delayMicroseconds(2); digitalWrite(T\_PIN, HIGH); delayMicroseconds(10); digitalWrite(T\_PIN, LOW);

int duration = pulseIn(E\_PIN, HIGH); return duration \* 0.034 / 2;

}

void loop() {

float distance = readDistanceCM(); Serial.print("Measured distance: "); Serial.println(distance); if(distance<=100){ PublishData(distance);

}

delay(1000);

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

}

}

void PublishData(float distance) {

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

/\*

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

\*/

bool status=true;

String payload = "{\"ALERT\_MESSAGE\":"; payload += status;

payload += "," "\"DISTANCE\":"; payload += distance;

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(".");

delay(500);

}

initManagedDevice();

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());

}

void initManagedDevice() {

if (client.subscribe(subscribetopic)) { Serial.println((subscribetopic)); Serial.println("subscribe to cmd OK");

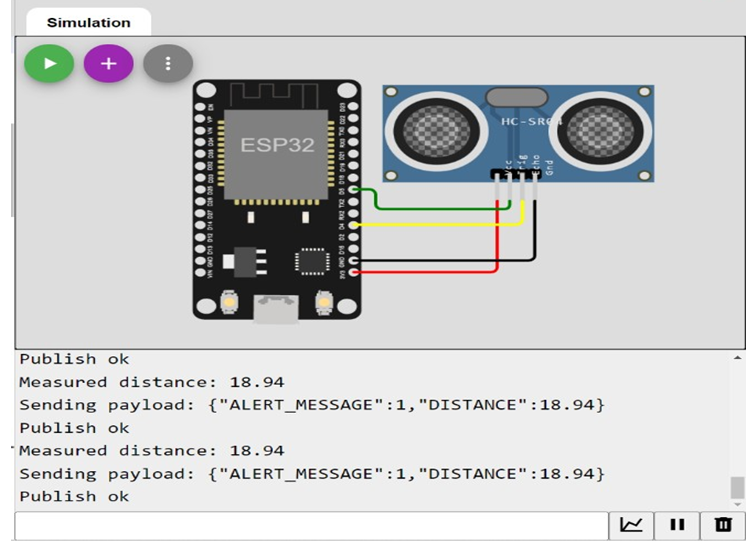
} else {

Serial.println("subscribe to cmd FAILED");

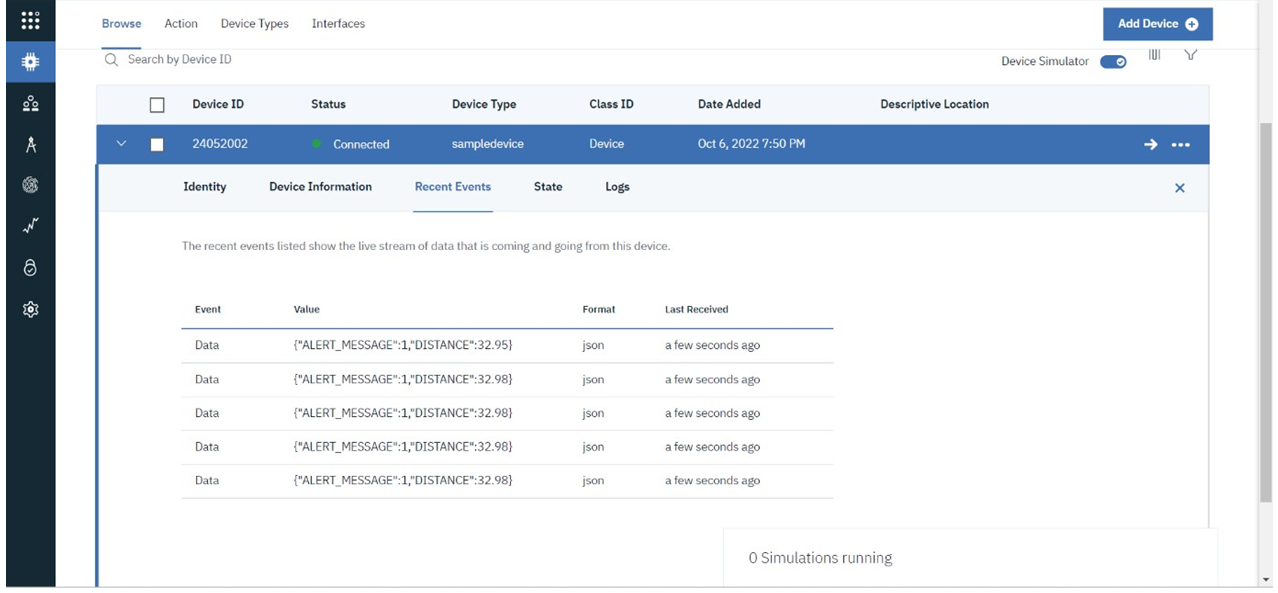
}

}

***Output:***

******

***IBM Cloud Image:***

****