ASSIGNMENT4

WOKWI PROGRAM

ASSIGNMNET DATE	23 OCTOBER 2022
STUDENT NAME	SABARIVASAN B
STUDENT ROLL NUMBER	110519106016
MAXIMUM NUMBER	2 MARKS
TEAM ID	PNT2022TMID36201

CODE :

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG "r1nl13"
#define DEVICE_TYPE "sabarivasan27"
#define DEVICE_ID "6016"
#define TOKEN "Z4sbS+lTfmaKWWn6uO"
String data3;


char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/sabarivasan/fmt/json";
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;
```

```
WiFiClientwifiClient;
PubSubClient client(server,1883,callback,wifiClient);

#define ECHO_PIN 12
#define TRIG_PIN 13
#define led 14

void setup() {
  // put your setup code here, to run once:
Serial.begin(115200);
pinMode(led, OUTPUT);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
wificonnect();
mqttconnect();
}
float readDistanceCM() {
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
  int duration=random(1,200);
```

```
  //Serial.println(duration);
  //duration = pulseIn(ECHO_PIN, HIGH);
  return duration ;
  //Serial.println(duration);

}

void loop() {
  float distance = readDistanceCM();
  //Serial.println(distance);

  bool isNearby = distance < 100;
digitalWrite(led, isNearby);

Serial.print("Measured distance: ");
Serial.println(distance);
  if(distance<100){
    PublishData2(distance);

}else{
    PublishData1(distance);

  }
  //PublishData(distance);
```

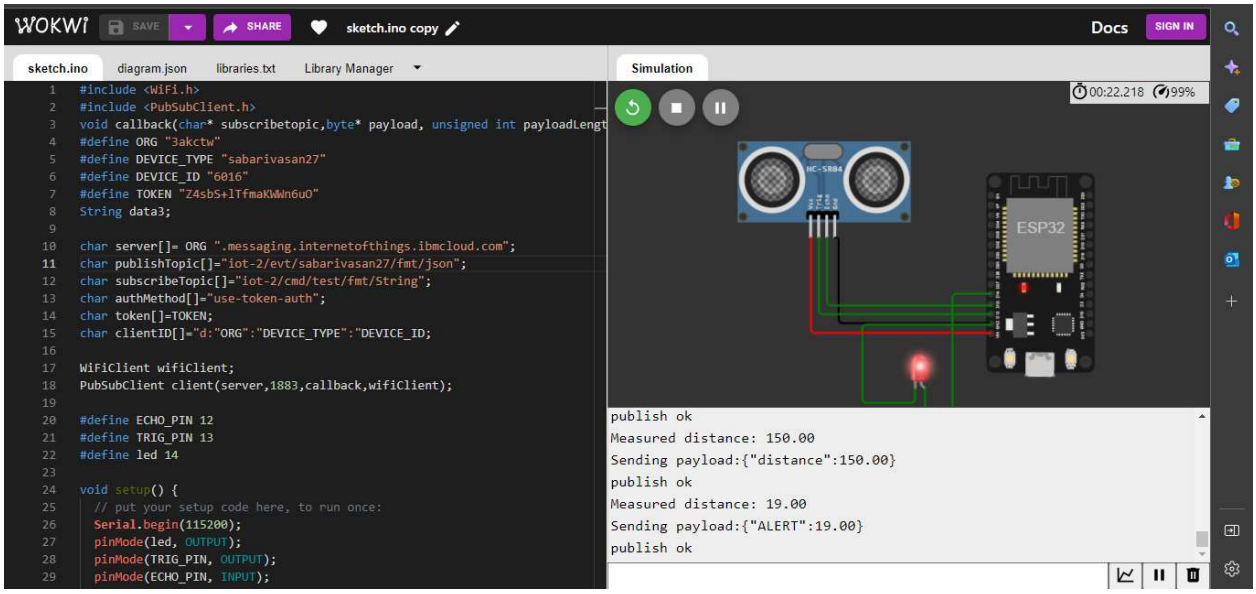
```
delay(1000);
  if(!client.loop()){
mqttconnect();
  }

  //delay(2000);
}
void PublishData1(float dist){
mqttconnect();
  String payload= "{\"distance\":";
  payload += dist;
  payload+="}";

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

  if(client.publish(publishTopic,(char*)payload.c_str())){
Serial.println("publish ok");
  } else{
Serial.println("publish failed");
  }
}
void PublishData2(float dist){
mqttconnect();
```

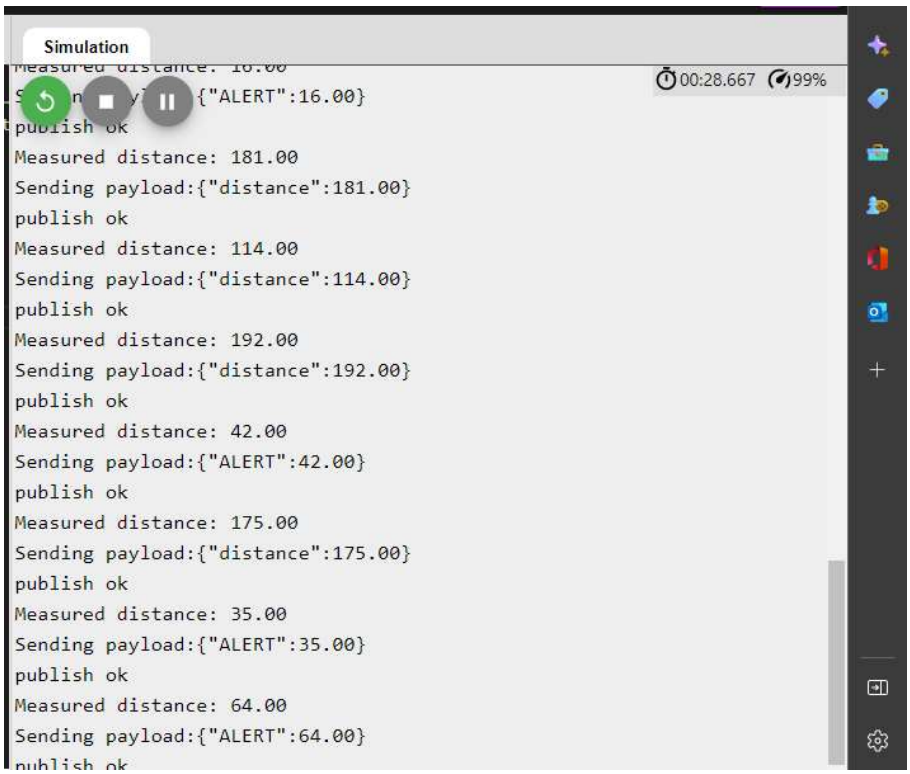
```
  String payload= "{\"ALERT\":";
  payload += dist;
  payload+="}";

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

  if(client.publish(publishTopic,(char*)payload.c_str())){
Serial.println("publish ok");
  } else{
Serial.println("publish failed");
  }
}
void mqttconnect(){
  if(!client.connected()){
Serial.print("Reconnecting to");
Serial.println(server);
while(!!!client.connect(clientID, authMethod, token)){
Serial.print(".");
delay(500);
    }
initManagedDevice();
Serial.println();
  }
```

```
}

void wificonnect(){
Serial.println();
Serial.print("Connecting to");

WiFi.begin("Wokwi-GUEST","",6);
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");
  }
```

```
}

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++){
    data3 += (char)payload[i];
  }
Serial.println("data:"+ data3);
  if(data3=="lighton"){
Serial.println(data3);
digitalWrite(led,HIGH);
}else{
Serial.println(data3);
digitalWrite(led,LOW);
  }
  data3="";
}
```

OUTPUT :


WOKWI
SAVE
SHARE
sketch.ino copy
Docs
SIGN IN
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:22.218
99%
ESP32
publish ok
Measured distance: 150.00
Sending payload:{"distance":150.00}
publish ok
Measured distance: 19.00
Sending payload:{"ALERT":19.00}
publish ok


Simulation
00:28.667
99%
{"ALERT":16.00}
publish ok
Measured distance: 181.00
Sending payload:{"distance":181.00}
publish ok
Measured distance: 114.00
Sending payload:{"distance":114.00}
publish ok
Measured distance: 192.00
Sending payload:{"distance":192.00}
publish ok
Measured distance: 42.00
Sending payload:{"ALERT":42.00}
publish ok
Measured distance: 175.00
Sending payload:{"distance":175.00}
publish ok
Measured distance: 35.00
Sending payload:{"ALERT":35.00}
publish ok
Measured distance: 64.00
Sending payload:{"ALERT":64.00}

IBM CLOUD OUTPUT :