ASSIGNMENT-4

Date	**24 October 2022**
Team ID	PNT2022TMID52857
Project Name	Project - IoT Based Safety Gadget for Child Safety Monitoring and Notification
Maximum Marks	**4 Marks**

Project Title: IoT Based Safety Gadget for Child Safety Monitoring and Notification

Team ID: PNT2022TMID52857

Team Members:

1. SAMUTHIRIKA.S 1904107
2. SHAFAHATH.S 1904110
3. SOORYA.R 2004208
4. TAMILARASAN .M 2004209

QUESTION:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

CODE:

```
#include <WiFi.h>//library for wifi
#include <WiFiClient.h>
#include <PubSubClient.h>//library for MQtt
// creating the instance by passing pin and typr of dht connected
 float distance;
#define sound_speed 0.034
 int trigpin=18;
 int echopin=19;
 int led=5;
 int LED=9;
 long duration;
 String message;// creating the instance by passing pin and typr of dht connected


void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

```
//-------credentials of IBM Accounts------


#define ORG "tmwrsv"//IBM ORGANITION ID

#define DEVICE_TYPE "iot_new"//Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "1110"//Device ID mentioned in ibm watson IOT Platform

#define TOKEN "BLQ_@ZfeL)6L@FM?kg"     //Token

String data3;

float h, t;


//-------- 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/command/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_TYPE ":" DEVICE_ID;//client id


//-------------------------------------

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32

{

  Serial.begin(115200);

pinMode(trigpin,OUTPUT);

pinMode(echopin,INPUT);

pinMode(led,OUTPUT);

  delay(10);

  Serial.println();

  wificonnect();

  mqttconnect();

}
```

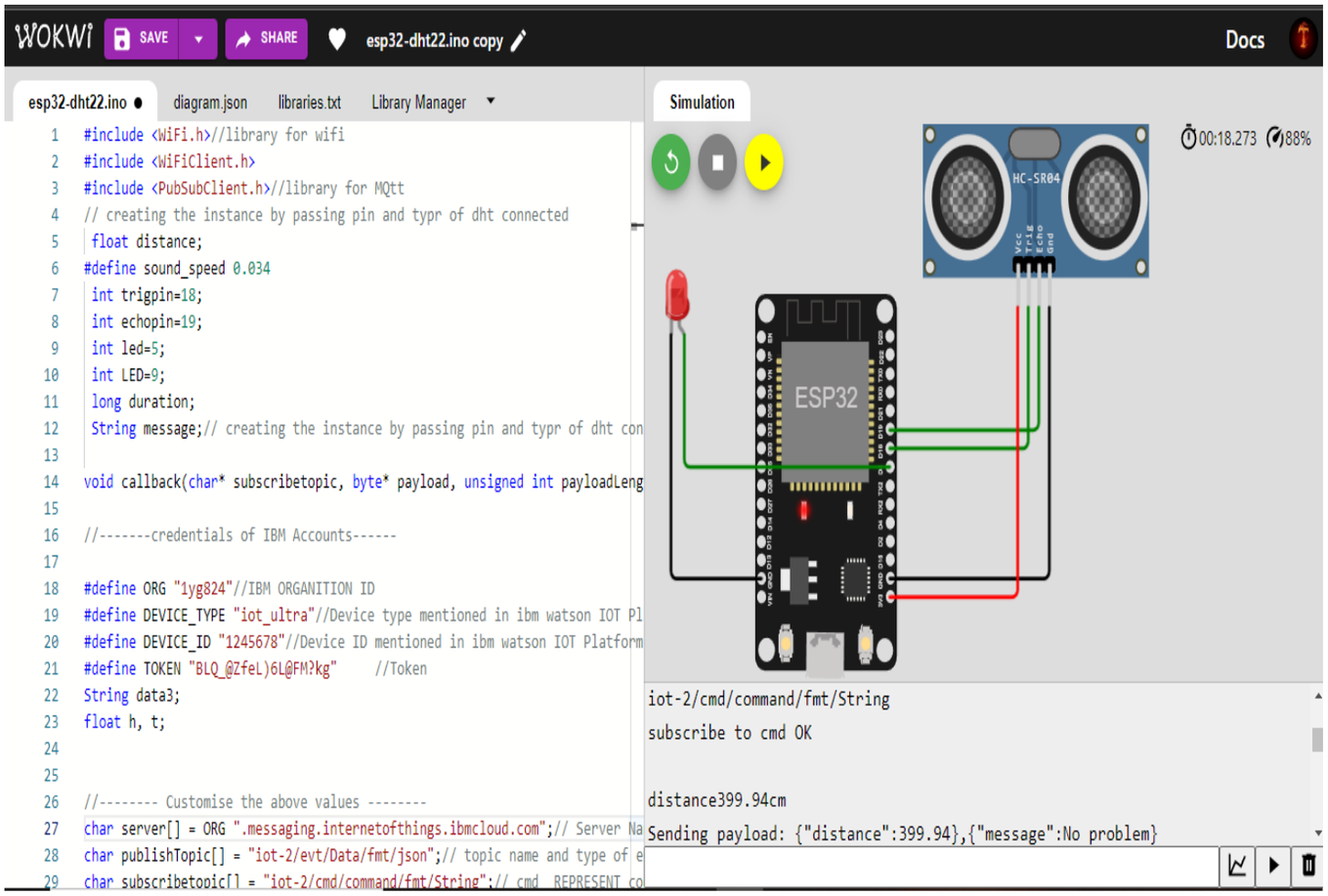
```
void loop()// Recursive Function
{

digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delay(1000);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
distance=duration*sound_speed/2;
Serial.println("distance"+String(distance)+"cm");
 if(distance<100)
 {
   message="Alert";
   digitalWrite(led,HIGH);
   } else
{
  message="No problem";
  digitalWrite(led,LOW);
  }
  delay(1000);
  PublishData(distance,message);
  // if (!client.loop()) {
  //   mqttconnect();
  // }
}


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

void PublishData(float d, String a) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"distance\":";
  payload += d; payload += "}";
```

```
  payload += "," "{\"message\":";
  payload += a;
  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");
  }
}

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++) {
    //Serial.print((char)payload[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="";
}
```

SCHEMATIC/CIRCUIT DIAGRAM:

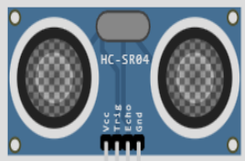


WOKWI SAVE SHARE esp32-dht22.ino copy Docs

esp32-dht22.ino ● diagram.json libraries.txt Library Manager

```
#define ORG "1yg824"//IBM ORGANITION ID
#define DEVICE_TYPE "iot_ultra"//Device type mentioned in ibm watson IOT Pl
#define DEVICE_ID "1245678"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "BLQ_@ZfeL)6L@FM?kg"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Na
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of e
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT co
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the pred
void setup()// configureing the ESP32
{
  Serial.begin(115200);
pinMode(trigpin,OUTPUT);
pinMode(echopin,INPUT);
pinMode(led,OUTPUT);
  delay(10);
  Serial.println();
```

Simulation



00:18.273 88%

```
iot-2/cmd/command/fmt/String
subscribe to cmd OK

distance399.94cm
Sending payload: {"distance":399.94},{"message":No problem}
Publish ok
distance399.96cm
Sending payload: {"distance":399.96},{"message":No problem}
Publish ok
distance399.92cm
Sending payload: {"distance":399.92},{"message":No problem}
Publish ok
distance399.94cm
Sending payload: {"distance":399.94},{"message":No problem}
Publish ok
distance399.94cm
```

IBM CLOUD OUTPUT: