Assignment -4

ESP32 to IBM IoT Watson platform

Assignment Date	04 November 2022
Student Name	SATHISH P
Student Roll Number	CITC1904109
Maximum Marks	2 Marks

Question:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cm 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.

Solution:

Wokwi – ESP32 Code

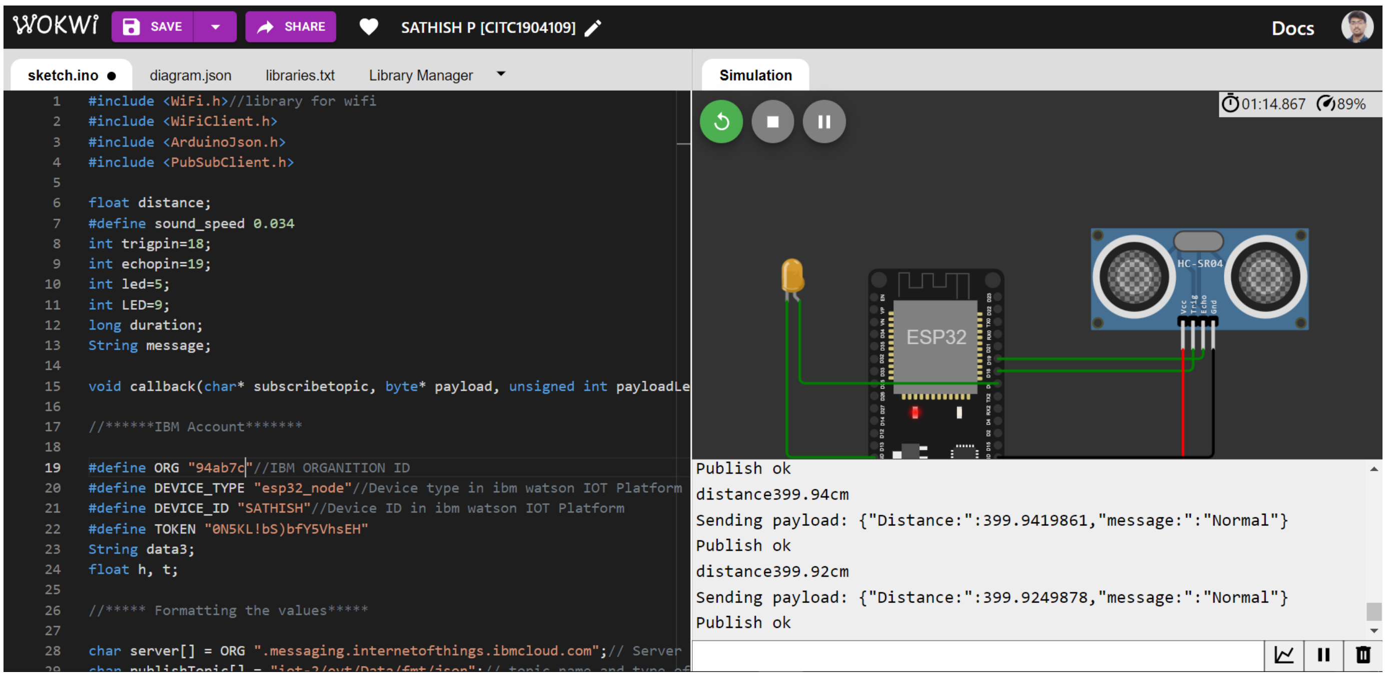
```
#include <WiFi.h>//library for wifi
#include <WiFiClient.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>
 float
distance;
#define sound_speed 0.034
int trigpin=18;  int
echopin=19; int led=5;
int LED=9;  long
duration;
String message;

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

//******IBM Account*******

#define ORG "94ab7c"//IBM ORGANITION ID
#define DEVICE_TYPE "esp32_node"//Device type in ibm watson IOT Platform
#define DEVICE_ID "SATHISH"//Device ID in ibm watson IOT Platform
#define TOKEN "0N5KL!bS)bfY5VhsEH"
String data3; float h, t;

//***** Formatting the 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()
{
  Serial.begin(115200);
pinMode(trigpin,OUTPUT);
pinMode(echopin,INPUT);
pinMode(led,OUTPUT);
delay(10);   Serial.println();
wificonnect();
mqttconnect();
}
void loop()
{  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="Normal";
digitalWrite(led,LOW);
  }
  delay(1000);
  PublishData(distance,message);
}

//*****Publish******
  void PublishData(float d, String
a)
{
  mqttconnect();

  //creating the String in in form JSon to update the data to ibm cloud
  DynamicJsonDocument doc(1024);
String payload;   doc["Distance:"]=d;
doc["message:"]=a;
serializeJson(doc, payload);
  Serial.print("Sending payload: ");
  Serial.println(payload);
```

```
   if (client.publish(publishTopic,(char*)
payload.c_str()))
  {
    Serial.println("Publish ok");// if upload sucessful
  }
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)
```

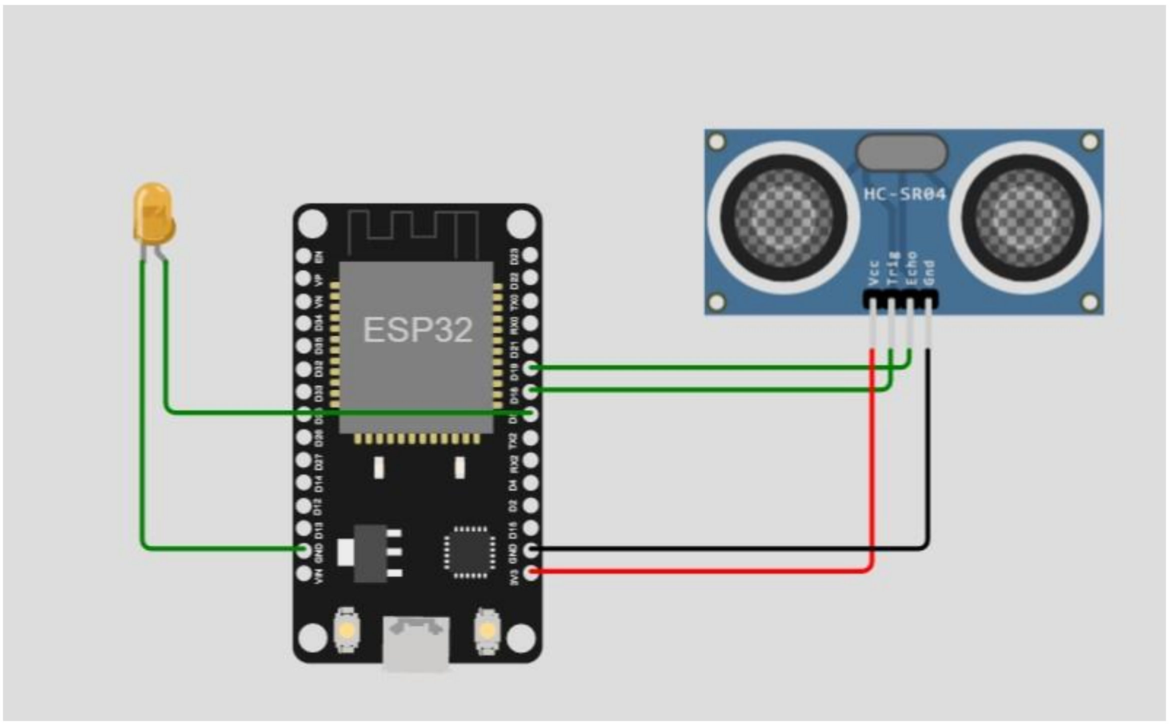
```
{

  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="";
}
```

Wokwi Circuit:

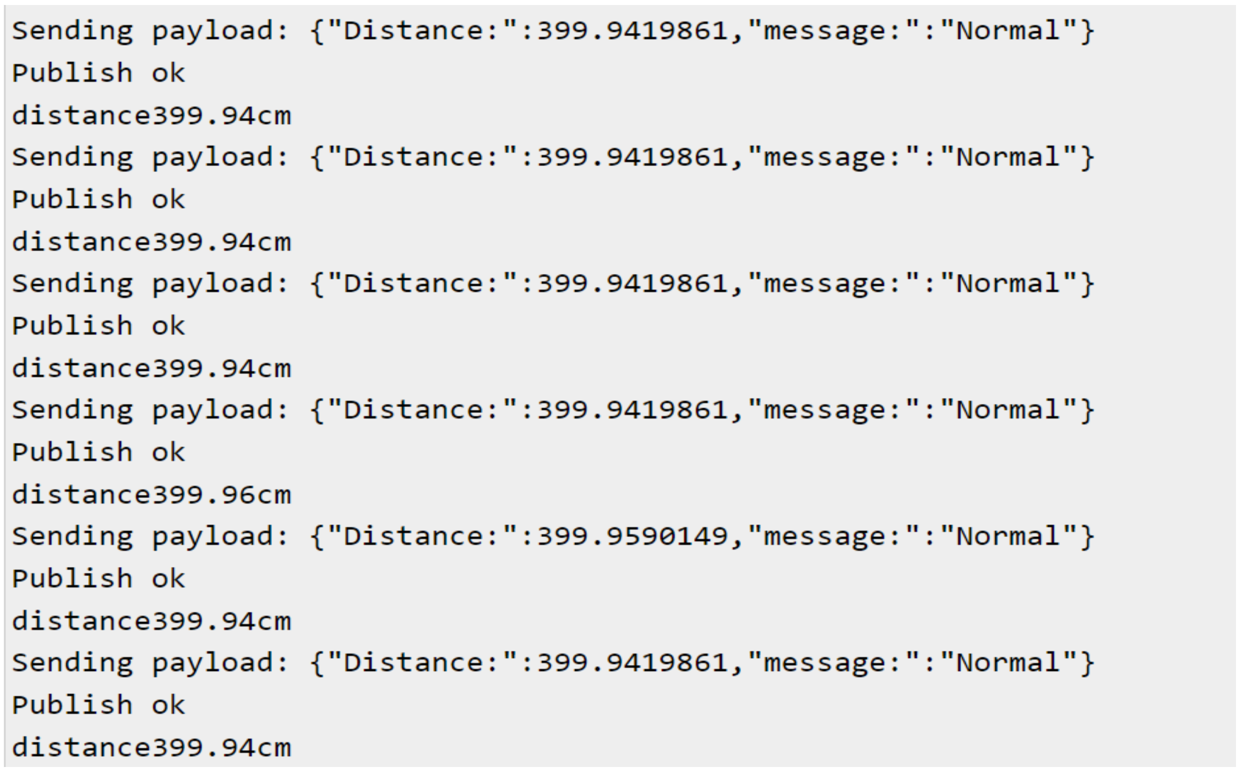


```
#include <WiFi.h>//library for wifi
#include <WiFiClient.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>

float distance;
#define sound_speed 0.034
int trigpin=18;
int echopin=19;
int led=5;
int LED=9;
long duration;
String message;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLe

//******IBM Account*******

#define ORG "94ab7c"//IBM ORGANITION ID
#define DEVICE_TYPE "esp32_node"//Device type in ibm watson IOT Platform
#define DEVICE_ID "SATHISH"//Device ID in ibm watson IOT Platform
#define TOKEN "0N5KL!bS)bfY5VhsEH"
String data3;
float h, t;

//***** Formatting the values*****

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
```

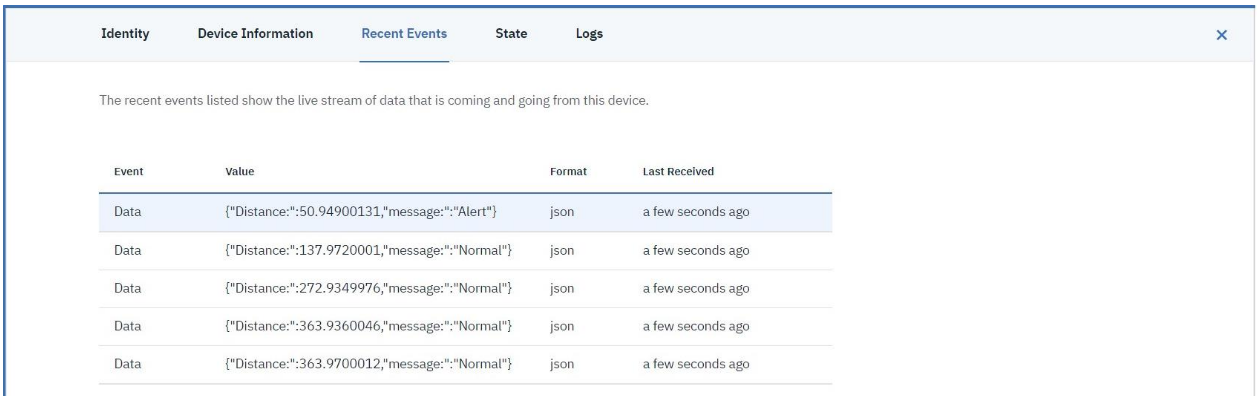
```
Publish ok
distance399.94cm
Sending payload: {"Distance:":399.9419861,"message:":"Normal"}
Publish ok
distance399.92cm
Sending payload: {"Distance:":399.9249878,"message:":"Normal"}
Publish ok
```



SIMULATION:

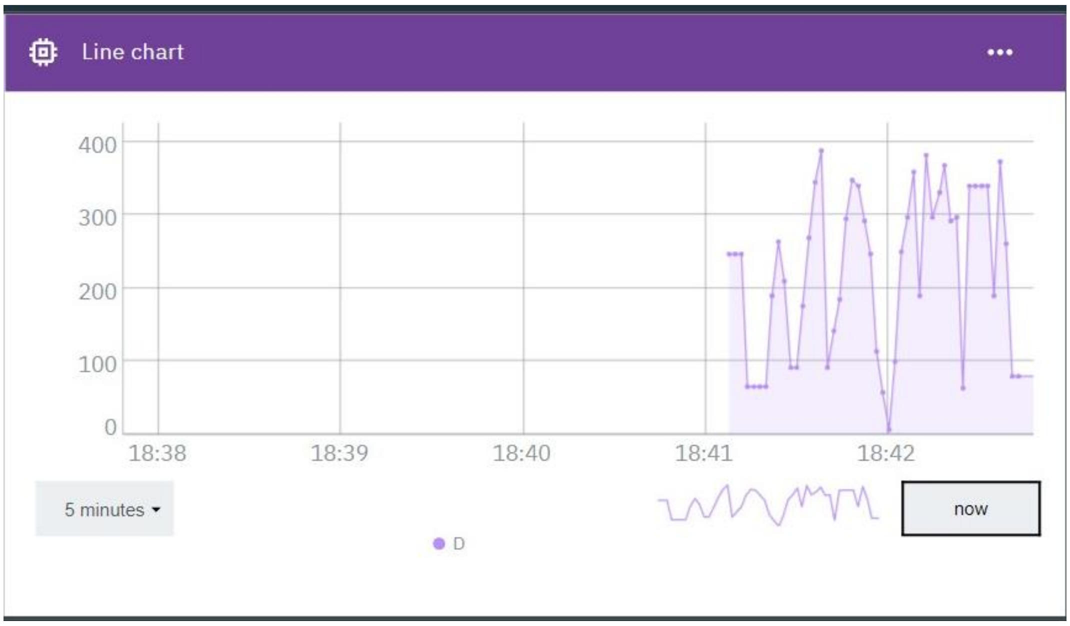


```
Sending payload: {"Distance:":399.9419861,"message:":"Normal"}
Publish ok
distance399.94cm
Sending payload: {"Distance:":399.9419861,"message:":"Normal"}
Publish ok
distance399.94cm
Sending payload: {"Distance:":399.9419861,"message:":"Normal"}
Publish ok
distance399.94cm
Sending payload: {"Distance:":399.9419861,"message:":"Normal"}
Publish ok
distance399.96cm
Sending payload: {"Distance:":399.9590149,"message:":"Normal"}
Publish ok
distance399.94cm
Sending payload: {"Distance:":399.9419861,"message:":"Normal"}
Publish ok
distance399.94cm
```



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

Event	Value	Format	Last Received
Data	{"Distance:":50.94900131,"message:":"Alert"}	json	a few seconds ago
Data	{"Distance:":137.9720001,"message:":"Normal"}	json	a few seconds ago
Data	{"Distance:":272.9349976,"message:":"Normal"}	json	a few seconds ago
Data	{"Distance:":363.9360046,"message:":"Normal"}	json	a few seconds ago
Data	{"Distance:":363.9700012,"message:":"Normal"}	json	a few seconds ago



WOKWI URL: https://wokwi.com/projects/348312487890780756