Assignment -4

ESP32 Programming with IBM Cloud

Assignment Date	25 October 2022
Student Name	YAMINI M
Student Roll Number	**2116191001110**
Maximum Marks	2 Marks

Question-1:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cms send “alert” to ibm cloud and display in device recent events.

Upload document with wokwi share link and images of ibm cloud.

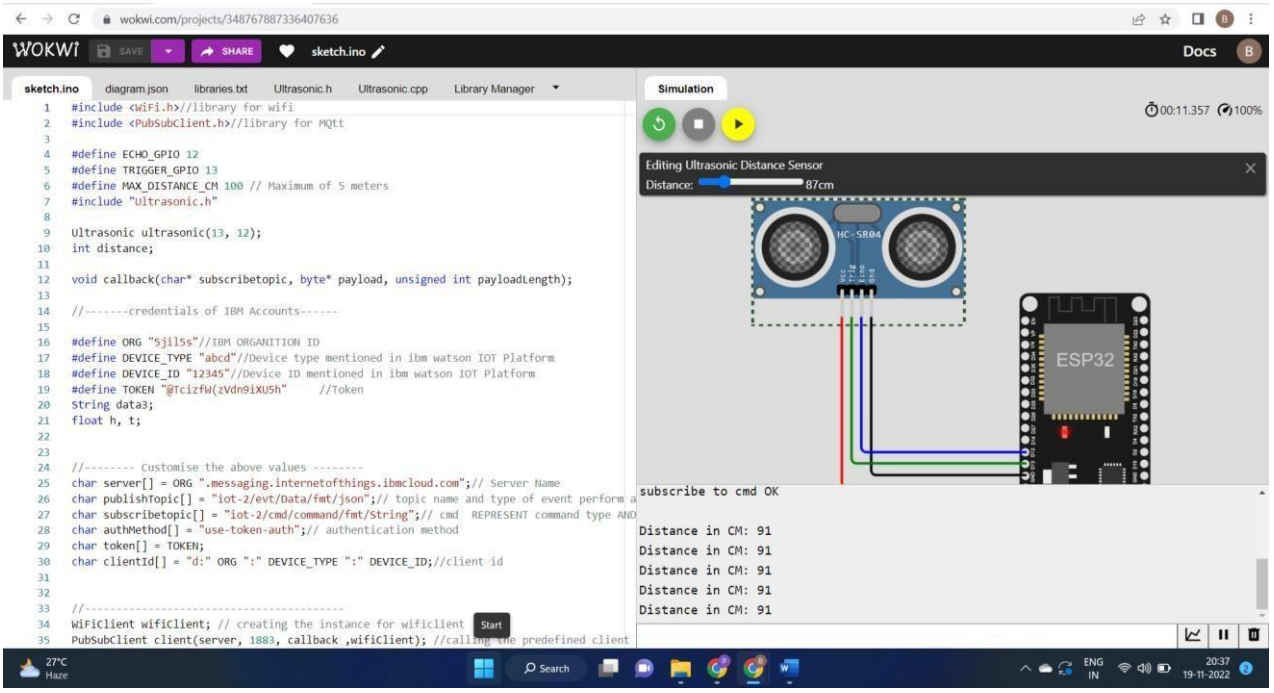
Solution:

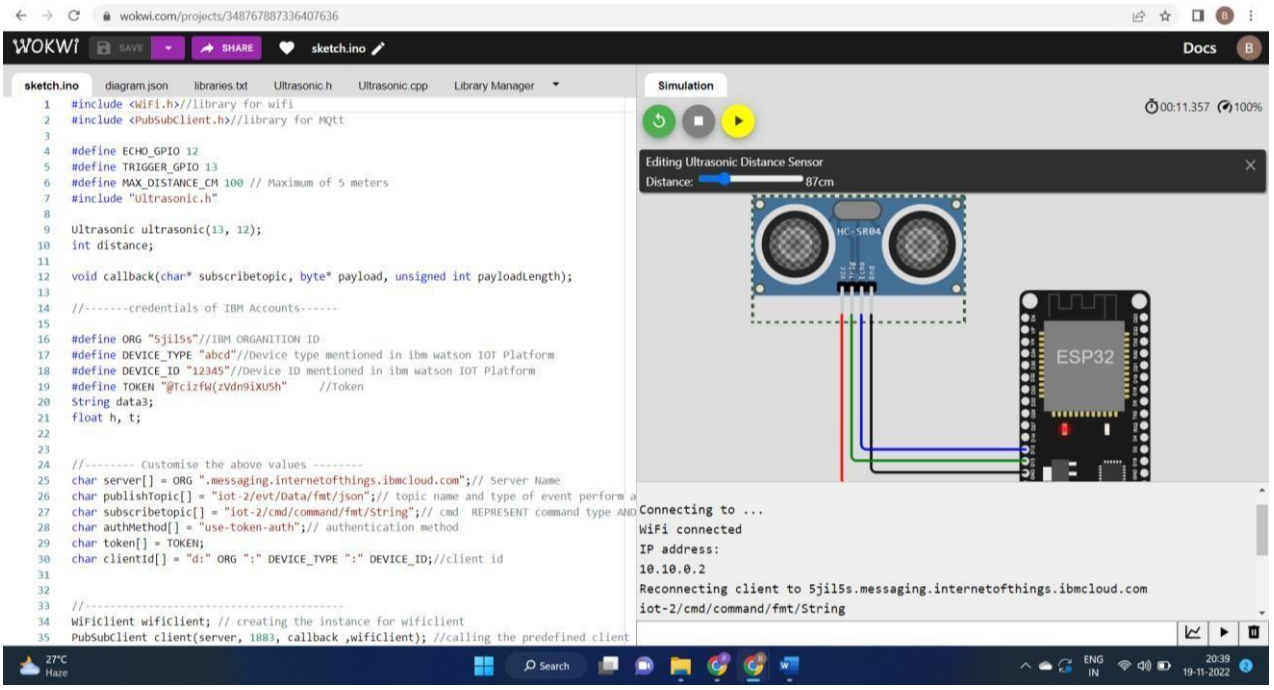
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

#define ECHO_GPIO 12
#define TRIGGER_GPIO 13
#define MAX_DISTANCE_CM 100 // Maximum of 5 meters
#include "Ultrasonic.h"

Ultrasonic ultrasonic(13, 12);
int distance;

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

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

#define ORG "5jil5s"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "@TcizfW(zVdn9iXU5h"      //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
```

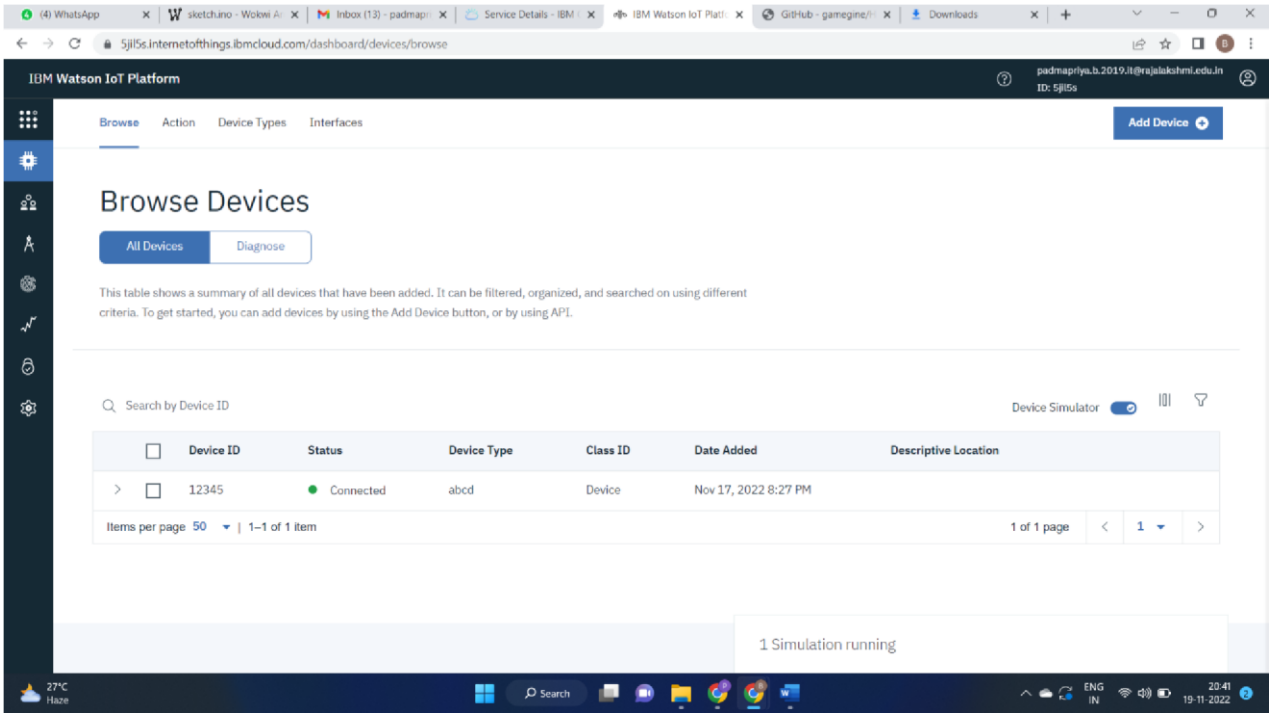
```
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type
AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method  char
token[] = TOKEN;  char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;//client id


//-------------------------------------
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);
delay(10);   Serial.println();
wificonnect();
mqttconnect();
}

void loop()// Recursive Function
{

  distance = ultrasonic.read(CM);   if(distance
< 100){
  Serial.print("Distance in CM: ");
  Serial.println(distance);
PublishData(distance);
delay(1000);   if (!client.loop())
{     mqttconnect();
  }

  }

  delay(1000);


}

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

void PublishData(float temp) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
```

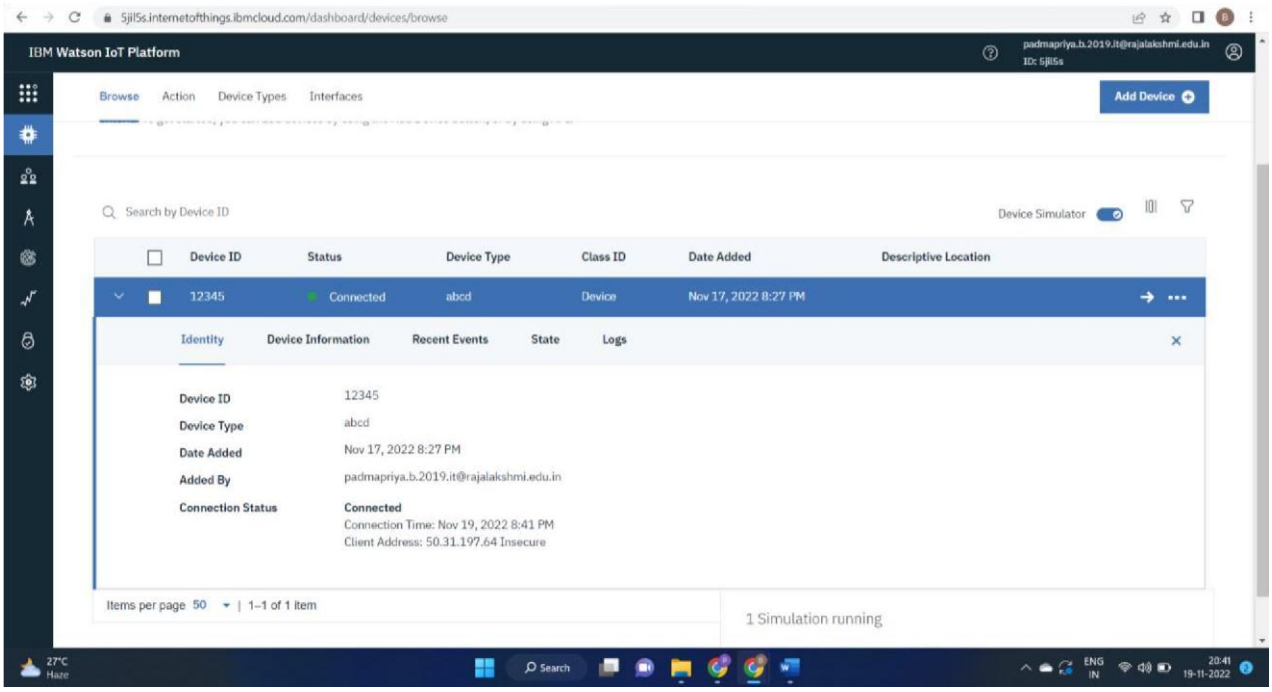
```
 */
 String payload = "{\"Alert Distance:\":";   payload
+= temp;   payload += "}";



}

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++) {
```

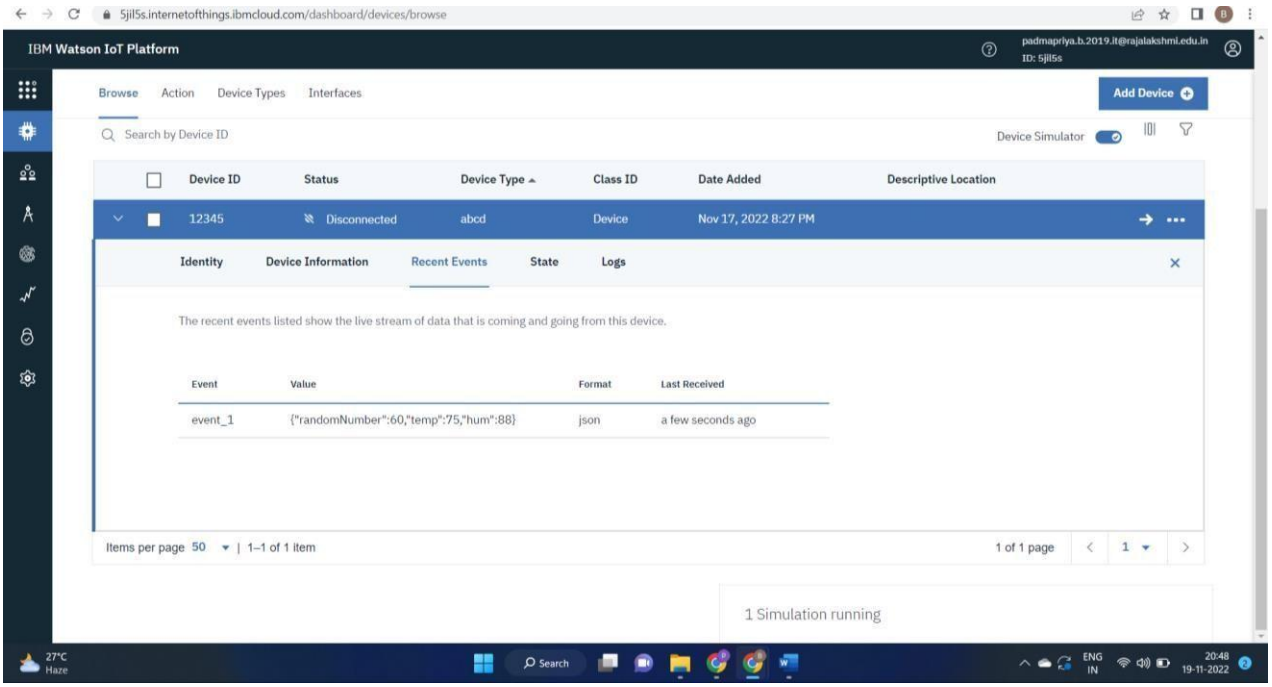
```
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);    if(data3=="lighton")
  {
Serial.println(data3);
  }   else
  {
Serial.println(data3);
  } data3="";
}
```




wokwi.com/projects/348767887336407636
WOKWI
SAVE
SHARE
sketch.ino
Docs
sketch.ino
diagram.json
libraries.txt
Ultrasonic.h
Ultrasonic.cpp
Library Manager
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define ECHO_GPIO 12
#define TRIGGER_GPIO 13
#define MAX_DISTANCE_CM 100 // Maximum of 5 meters
#include "Ultrasonic.h"
Ultrasonic ultrasonic(13, 12);
int distance;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------
#define ORG "5jil5s"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "@TcizfW(zVdn9iXU5h" //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 a
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND
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
Simulation
00:11.357
100%
Editing Ultrasonic Distance Sensor
Distance:
87cm
HC-SR04
ESP32
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 5jil5s.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
27°C
Haze
Search
ENG
IN
20:39
19-11-2022


5jil5s.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
padmapriya.b.2019.it@rajalakshmi.edu.in
ID: 5jil5s
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345
Connected
abcd
Device
Nov 17, 2022 8:27 PM
Items per page 50
1–1 of 1 item
1 of 1 page
1
1 Simulation running
27°C
Haze
Search
ENG
IN
20:41
19-11-2022





Wokwi share link:

https://wokwi.com/projects/348767887336407636