Assignment -4

ESP32 Programming

Assignment Date	1 November 2022
Student Name	Jeen Liberta J
Student Roll Number	962819106019
Maximum Marks	2 Marks

Assignment 4:

Write code and connections in wokwi for the ultrasonic sensor.

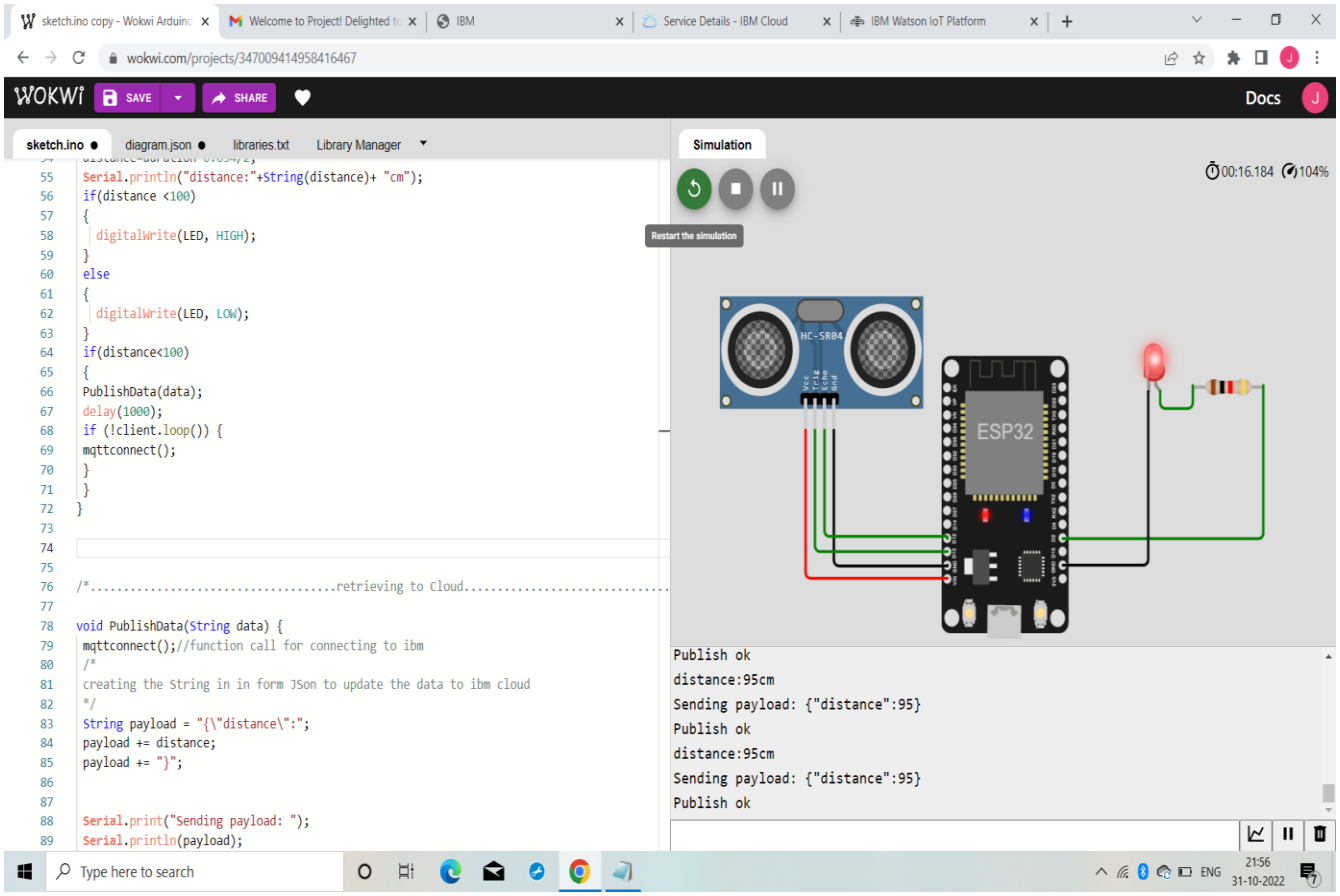
Whenever the distance is less than 100 cms send an “alert” to IBM cloud and display in the device recent events.

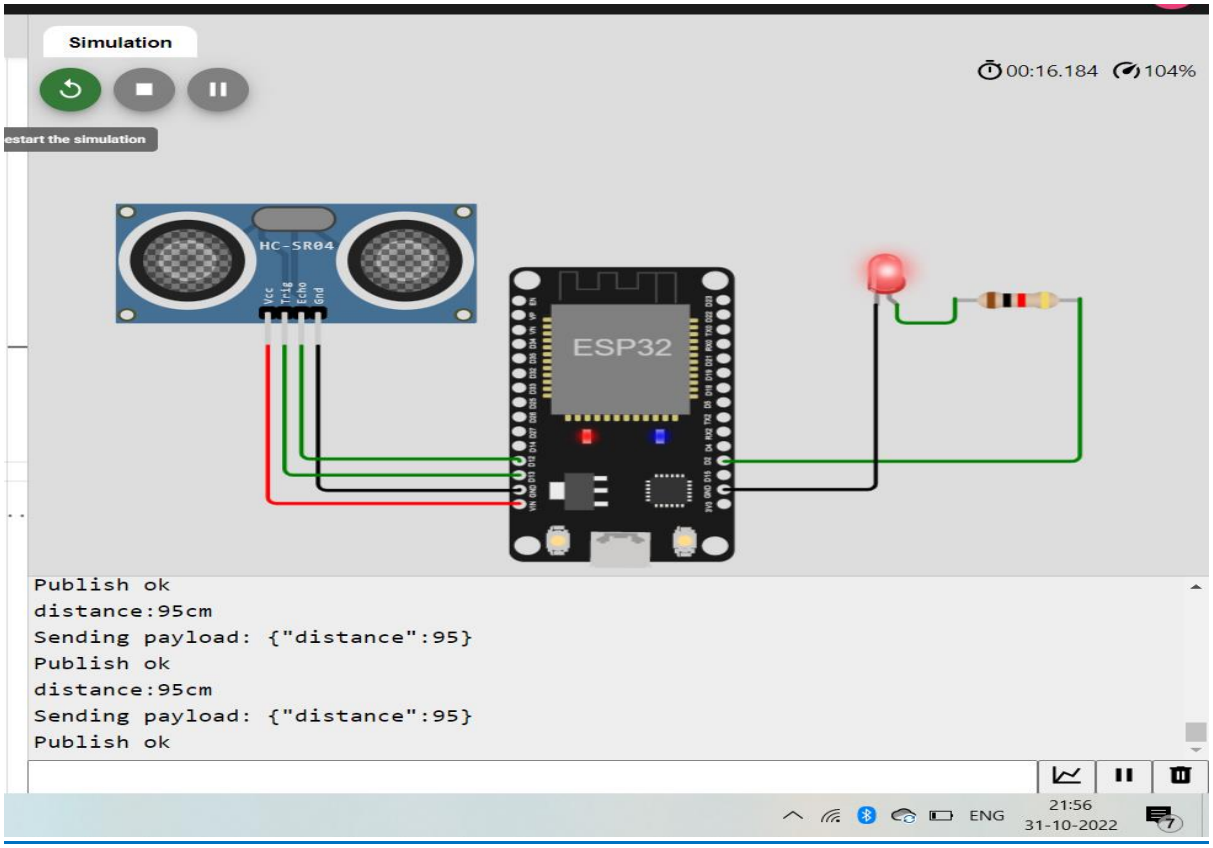
Upload document with wokwi share link and images of IBM cloud.

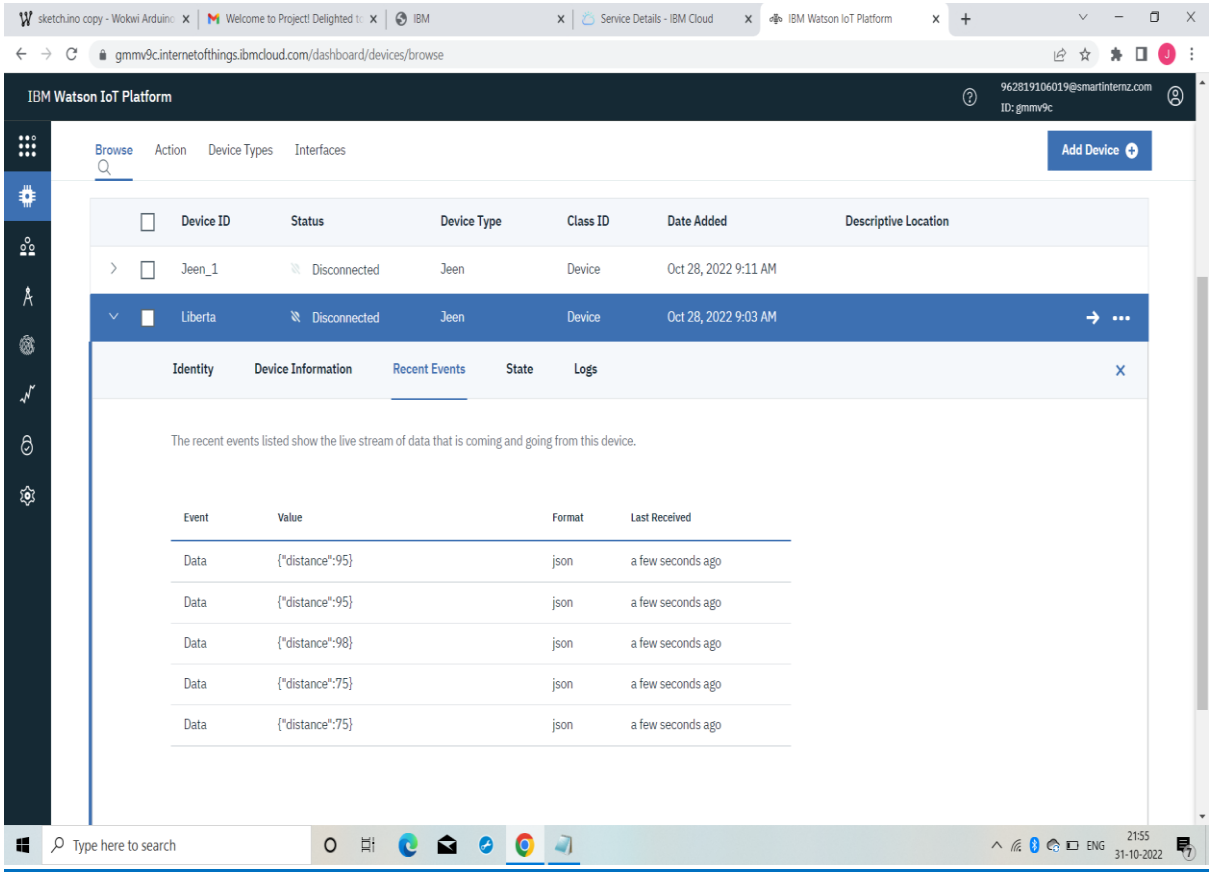
My Completed Assignment Wokwi Share Link:

https://wokwi.com/projects/347009414958416467

Circuit diagram:




Simulation
00:16.184
104%
estart the simulation
HC-SR04
Vcc
Trig
Echo
Gnd
ESP32
Publish ok
distance:95cm
Sending payload: {"distance":95}
Publish ok
distance:95cm
Sending payload: {"distance":95}
Publish ok
ENG
21:56
31-10-2022


gmmv9c.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
962819106019@smartinternz.com
ID: gmmv9c
Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
Jeen_1
Disconnected
Jeen
Device
Oct 28, 2022 9:11 AM
Liberta
Disconnected
Jeen
Device
Oct 28, 2022 9:03 AM
Identity
Device Information
Recent Events
State
Logs
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":95}
json
a few seconds ago
Data
{"distance":95}
json
a few seconds ago
Data
{"distance":98}
json
a few seconds ago
Data
{"distance":75}
json
a few seconds ago
Data
{"distance":75}
json
a few seconds ago
Type here to search
ENG
21:55
31-10-2022

Code:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define LED 2
#define echoPin 12
#define trigPin 13

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

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

#define ORG "gmmv9c"//IBM ORGANITION ID
#define DEVICE_TYPE "Jeen"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Liberta"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "UR68V3+R2b3&fz5gJq" //Token
String data3;
long duration;
int distance;
String data;


//-------- 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);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH );
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration=pulseIn(echoPin, HIGH);
 distance=duration*0.034/2;
 Serial.println("distance:"+String(distance)+ "cm");
 if(distance <100)
 {
   digitalWrite(LED, HIGH);
 }
 else
 {
   digitalWrite(LED, LOW);
 }
 if(distance<100)
 {
 PublishData(data);
 delay(1000);
 if (!client.loop()) {
 mqttconnect();
 }
 }
}



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

void PublishData(String data) {
 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 += 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");
```

```
 }
}

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);
data3="";


}
```