ASSIGNMENT 4
MAHENDRA INSTITUTE OF TECHNOLOGY

Date	28 October2022
Team ID	PNT2022TMID17220
Project Name	Signs with smart connectivity for better road safety
Maximum Marks	2 Marks

Question:

Write code and connections in wowki for ultrasonic sensor. Whenever distance is less than 100cms send "alert" to IBM cloud anddisplay in device recent events.

Program Code:

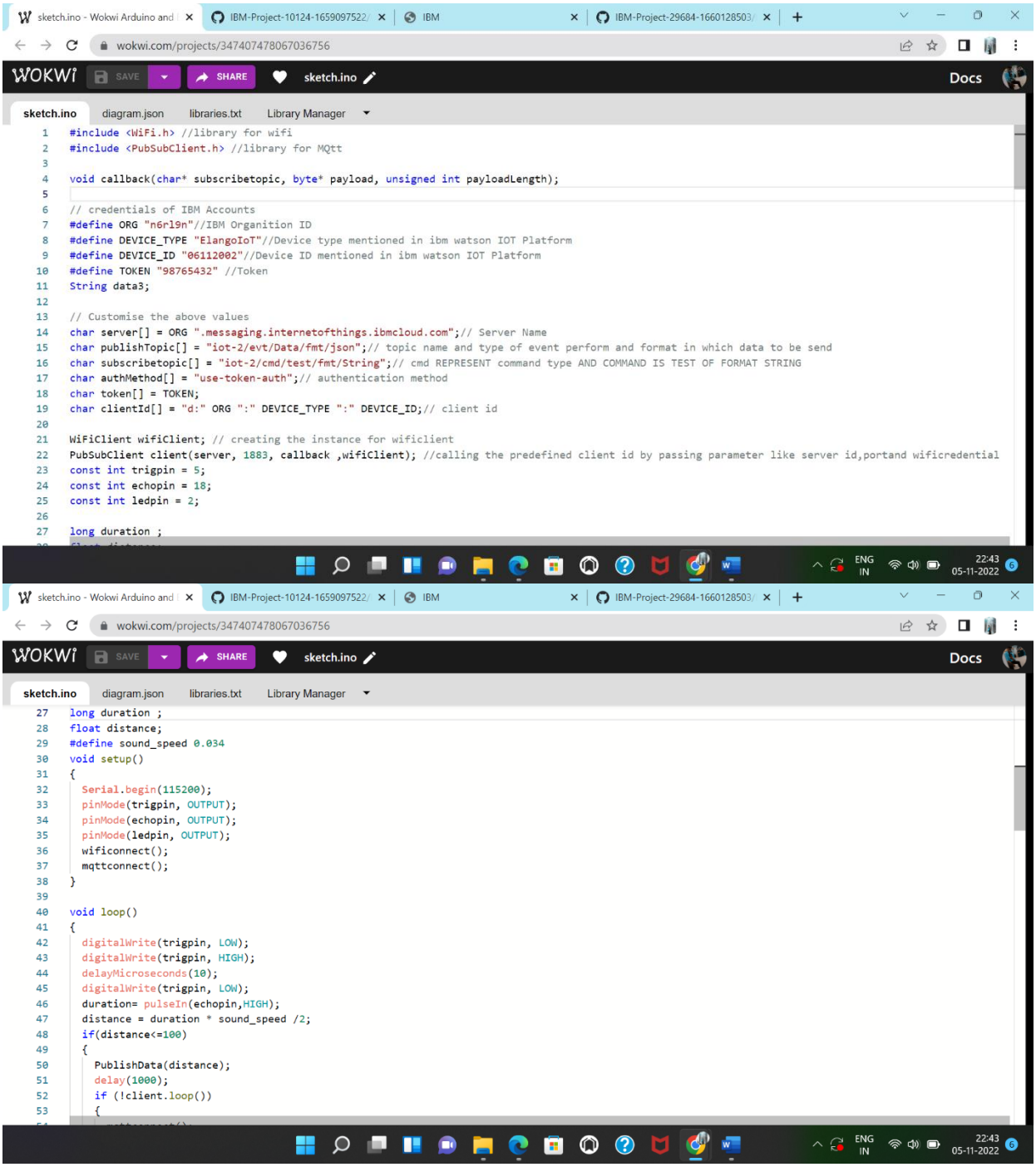
```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQtt

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

// credentials of IBM Accounts
#define ORG "n6rl9n"//IBM Organition ID
#define DEVICE_TYPE "ElangoIoT"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "06112002"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "98765432" //Token
String data3;

// 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/test/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
```

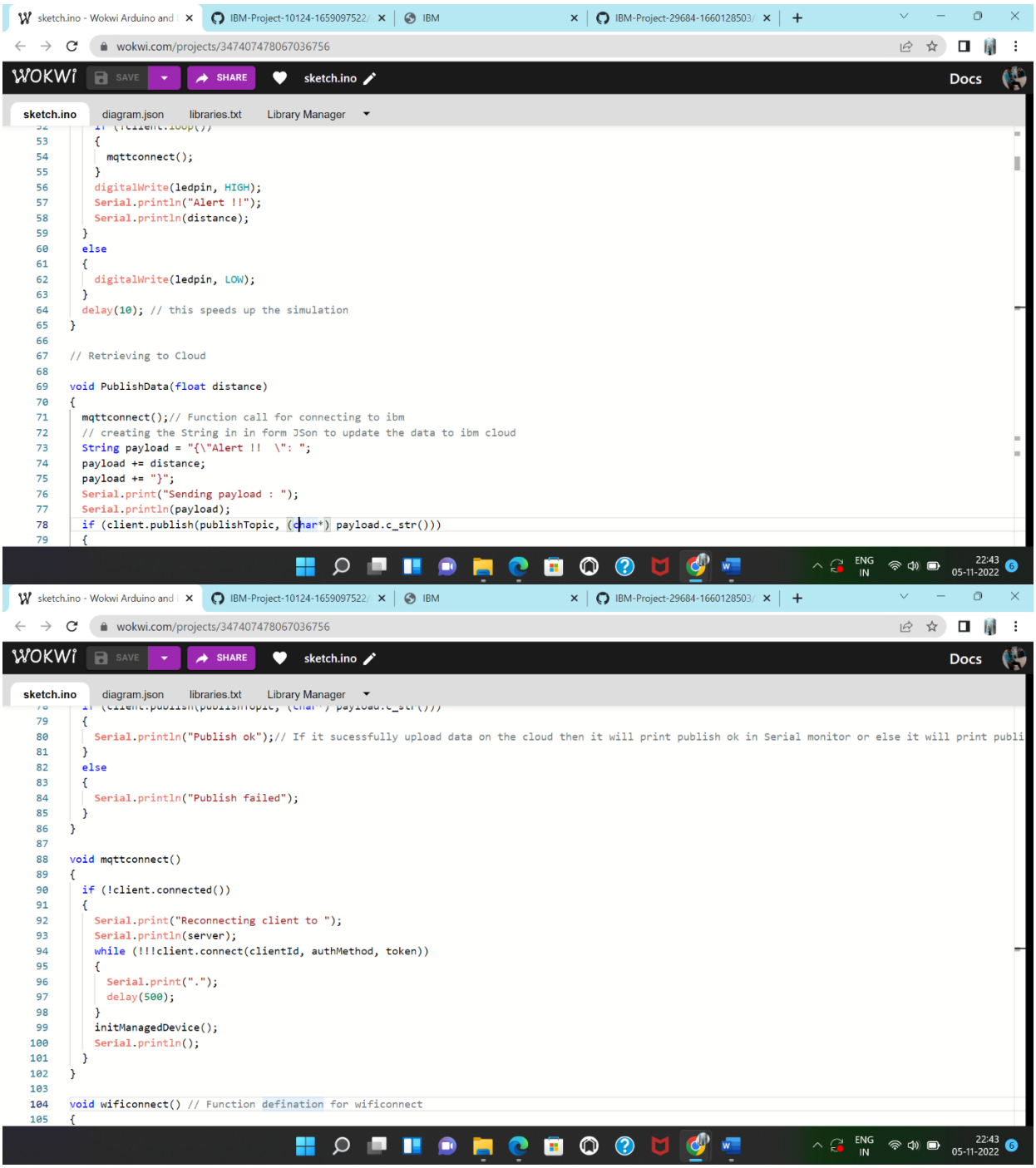
```
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;

long duration ;
float distance;
#define sound_speed 0.034
void setup()
{
  Serial.begin(115200);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, OUTPUT);
  pinMode(ledpin, OUTPUT);
  wificonnect();
  mqttconnect();
}

void loop()
{
  digitalWrite(trigpin, LOW);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);
  duration= pulseIn(echopin,HIGH);
  distance = duration * sound_speed /2;
  if(distance<=100)
  {
    PublishData(distance);
    delay(1000);
    if (!client.loop())
    {
      mqttconnect();
    }
    digitalWrite(ledpin, HIGH);
    Serial.println("Alert !!");
    Serial.println(distance);
  }
  else
  {
    digitalWrite(ledpin, LOW);
  }
  delay(10); // this speeds up the simulation
}

// Retrieving to Cloud

void PublishData(float distance)
{
  mqttconnect();// Function call for connecting to ibm
```

```
  // creating the String in in form JSon to update the data to ibm cloud
  String payload = "{\"Alert !!  \": ";
  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());
```

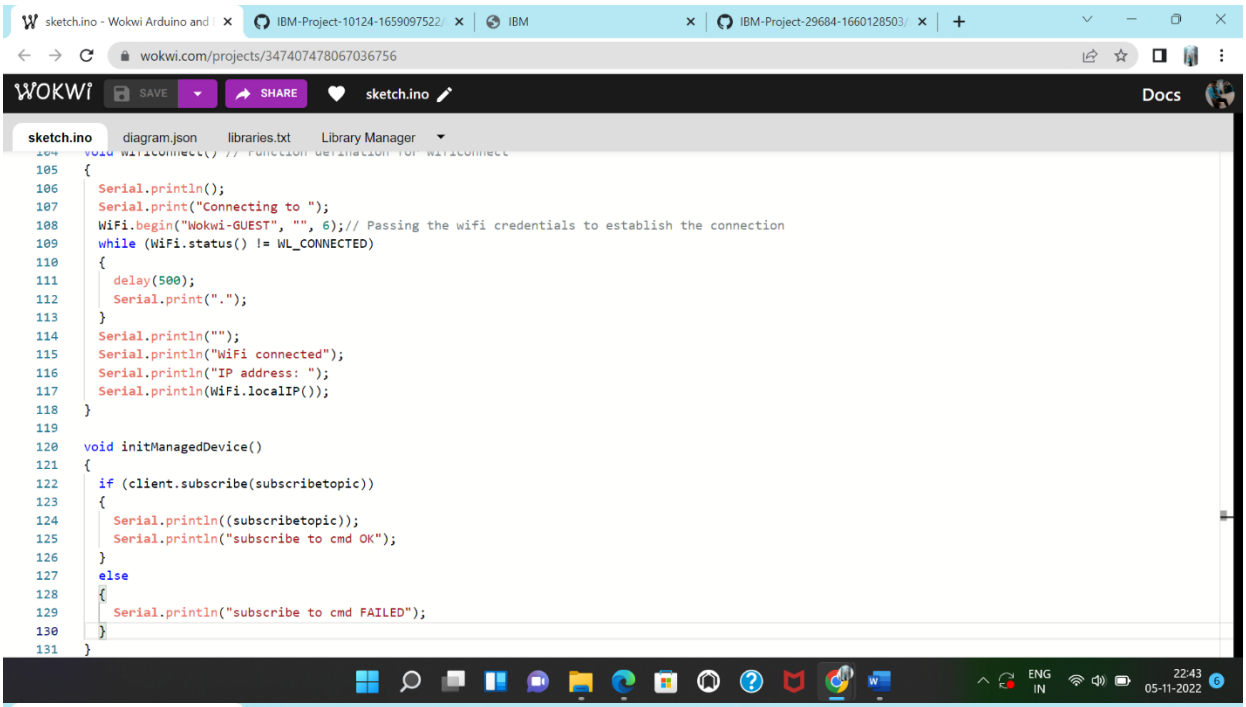
```
}

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);
  if(data3=="lighton")
  {
      Serial.println(data3);
  }
  else
  {
      Serial.println(data3);
  }
data3="";
}
```

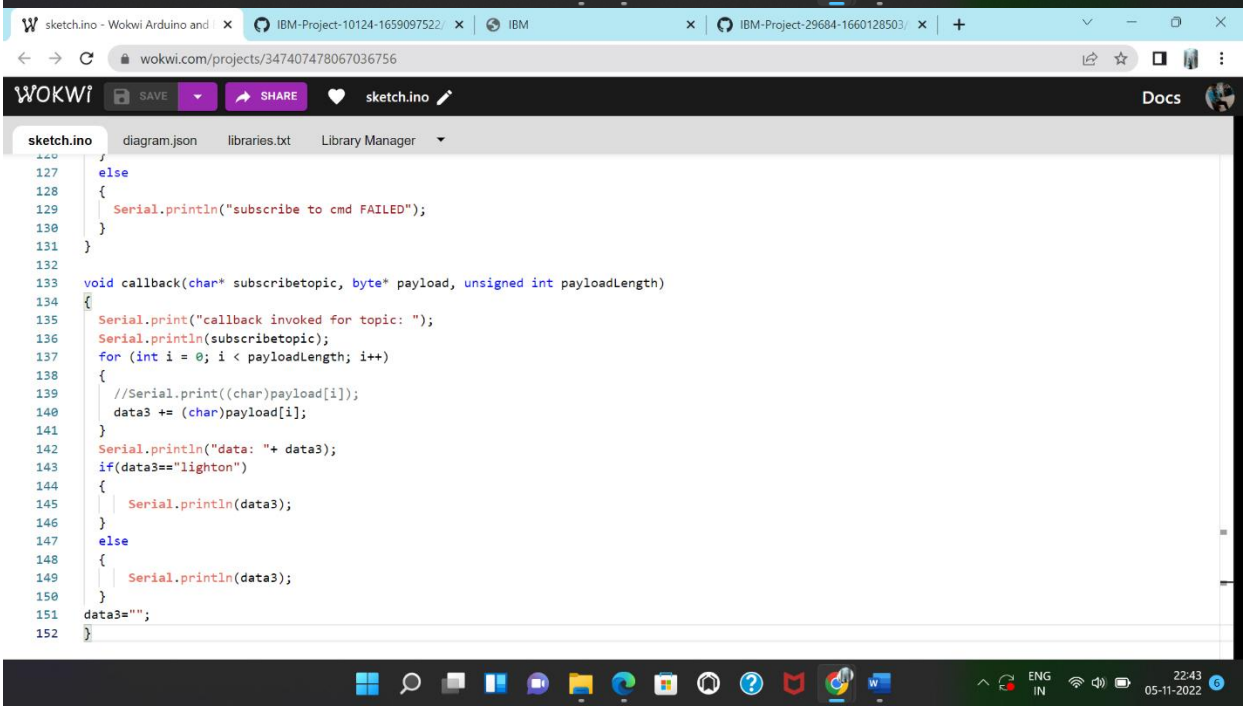
OUTPUT:

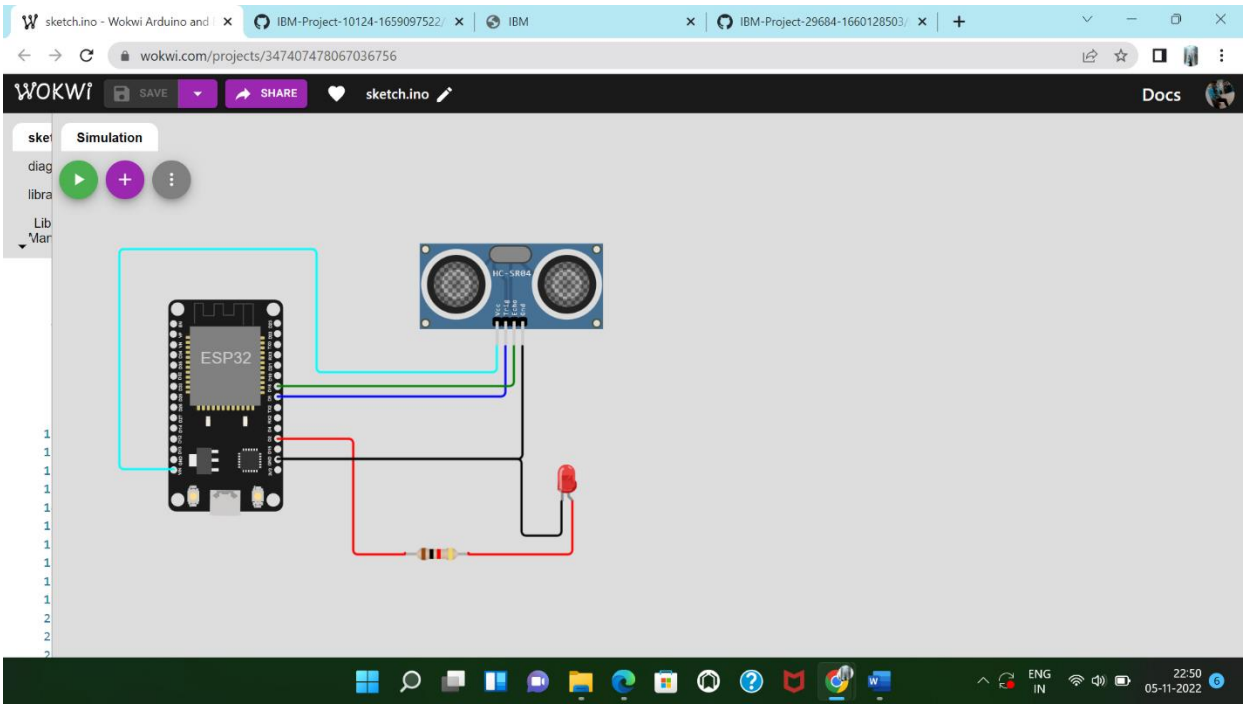

sketch.ino - Wokwi Arduino and
IBM-Project-10124-1659097522/
IBM
IBM-Project-29684-1660128503/
wokwi.com/projects/347407478067036756
WOKWI
SAVE
SHARE
sketch.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQtt
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
// credentials of IBM Accounts
#define ORG "n6rl9n"//IBM Organition ID
#define DEVICE_TYPE "ElangoIoT"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "06112002"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "98765432" //Token
String data3;
// 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/test/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
const int trigpin = 5;
const int echopin = 18;
const int ledpin = 2;
long duration ;
ENG
IN
22:43
05-11-2022
sketch.ino - Wokwi Arduino and
IBM-Project-10124-1659097522/
IBM
IBM-Project-29684-1660128503/
wokwi.com/projects/347407478067036756
WOKWI
SAVE
SHARE
sketch.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
long duration ;
float distance;
#define sound_speed 0.034
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, OUTPUT);
pinMode(ledpin, OUTPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigpin, LOW);
digitalWrite(trigpin, HIGH);
delayMicroseconds(10);
digitalWrite(trigpin, LOW);
duration= pulseIn(echopin,HIGH);
distance = duration * sound_speed /2;
if(distance<=100)
{
PublishData(distance);
delay(1000);
if (!client.loop())
{
ENG
IN
22:43
05-11-2022



```
    {
      mqttconnect();
    }
    digitalWrite(ledpin, HIGH);
    Serial.println("Alert !!");
    Serial.println(distance);
  }
  else
  {
    digitalWrite(ledpin, LOW);
  }
  delay(10); // this speeds up the simulation
}

// Retrieving to Cloud

void PublishData(float distance)
{
  mqttconnect();// Function call for connecting to ibm
  // creating the String in in form JSon to update the data to ibm cloud
  String payload = "{\"Alert !!  \": ";
  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 publi
  }
  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
{
```


sketch.ino - Wokwi Arduino and
IBM-Project-10124-1659097522/
IBM
IBM-Project-29684-1660128503/
wokwi.com/projects/347407478067036756
WOKWI
SAVE
SHARE
sketch.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
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");
}
}
ENG
IN
22:43
05-11-2022


sketch.ino - Wokwi Arduino and
IBM-Project-10124-1659097522/
IBM
IBM-Project-29684-1660128503/
wokwi.com/projects/347407478067036756
WOKWI
SAVE
SHARE
sketch.ino
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
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="";
}
ENG
IN
22:43
05-11-2022



Referral Link: https://wokwi.com/projects/347407478067036756