SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITIAN CITIES

ASSIGNMENT-4

TEAM ID : PNT2022TMID04391
NAME : MOHAMED ROSHAN M
ROLL NO : 737819CSR108

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

CODE:

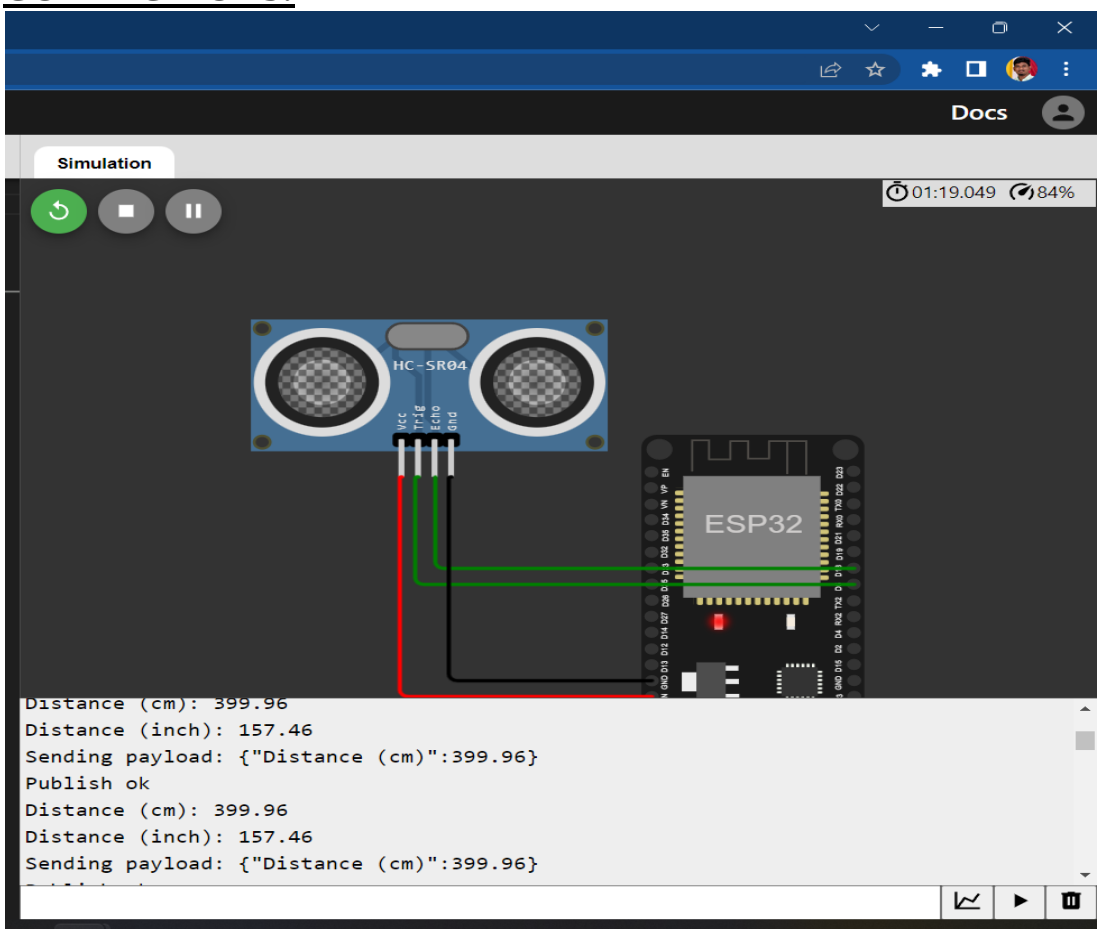
```
#include <WiFi.h>
#include<WiFiClient.h>
#include<PubSubClient.h>
const int trigPin = 5; const int echoPin = 18;
//define sound speed in cm/uS
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701
long duration;
float distanceCm;
float distanceInch;

void callback(char* subscribetopic, byte* payload, unsigned intpayloadLength);
//-------credentials ofIBMAccounts------

#define ORG "3defta"
#define DEVICE_TYPE "hariprasath"//Device type mentioned in ibm watson IOTPlatform
#define DEVICE_ID "12345"//Device ID mentioned in ibmwatson IOTPlatform
#define TOKEN "CpL-H1C-Pt4i9iM-F5" //Token
String data3;

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 commandtype AND COMMAND IS TEST OF FORMAT STRING
char authMethod[]="use-token-auth";

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);
```

```
void setup() {
Serial.begin(115200); // Starts the serial communication
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.println();
wifiConnect();
mqttConnect();

}

void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculate the distance
distanceCm = duration * SOUND_SPEED/2;

// Convert to inches
distanceInch = distanceCm * CM_TO_INCH;

// Prints the distance in the Serial Monitor
Serial.print("Distance (cm): ");
Serial.println(distanceCm);
Serial.print("Distance (inch): ");
Serial.println(distanceInch);

PublishData(distanceCm);
delay(1000);
if (!client.loop()) {
  mqttConnect();
}
}


void PublishData(float Cm) {

mqttConnect();//functioncall for connecting to ibm
/*
creating the String in in form JSon to update the data to ibm cloud
*/
String payload = "{\"Distance (cm)\":";
payload += Cm;
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 cloudthen it will print publish
ok in Serial monitor or else it will print publishfailed
}
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 forwificonnect
{
Serial.println();
Serial.print("Connecting to ");

WiFi.begin("Wokwi-GUEST", "", 6);//passing thewificredentials toestablishthe connection
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

Serial.println("IPaddress:");
Serial.println(WiFi.localIP());
}


void initManagedDevice()
{
if (client.subscribe(subscribetopic))
{
Serial.println((subscribetopic));
Serial.println("subscribetocmd OK");
} else
{
Serial.println("subscribetocmd 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];
}
}
```

CONNECTIONS:

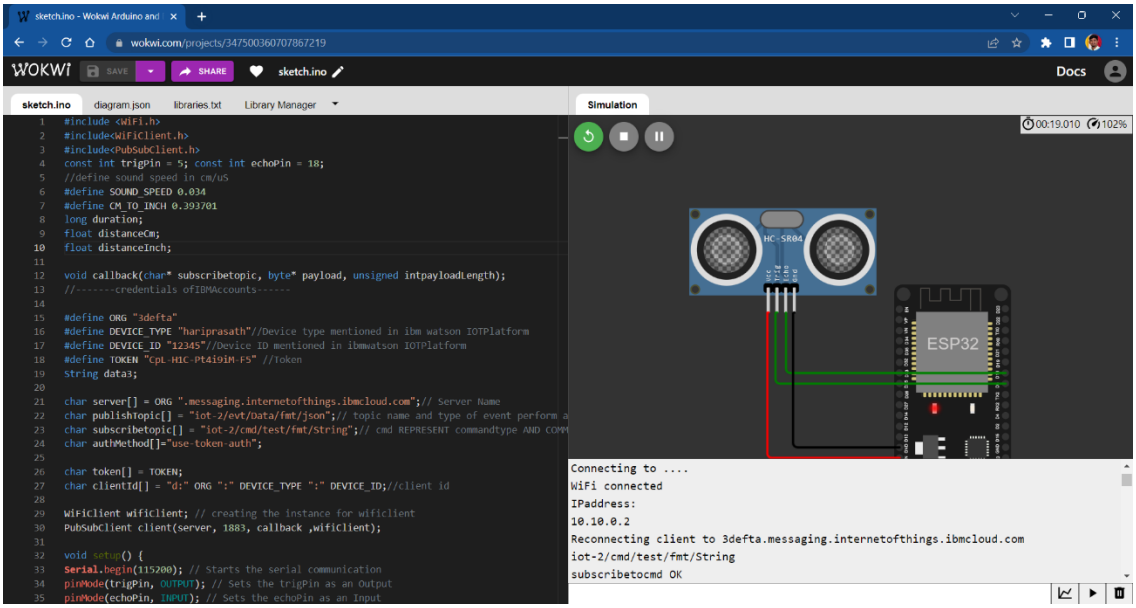


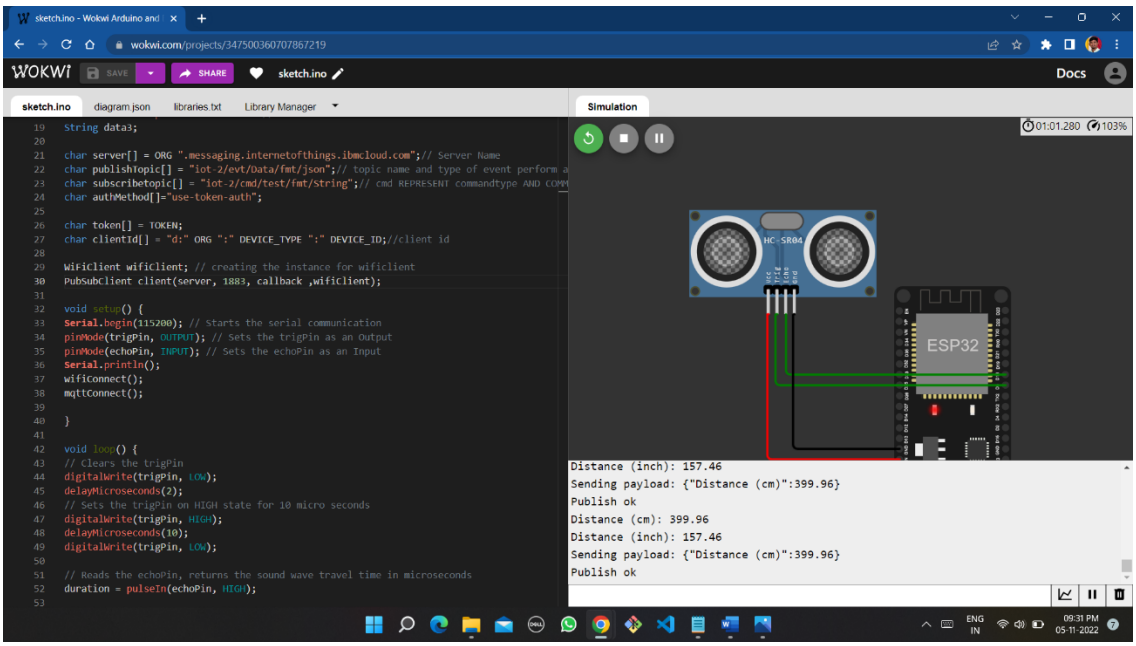
WOKWI LINK:

Code link:

https://wokwi.com/projects/347500360707867219

OUTPUT:





IBM Watson IOT platform connected: