REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

SUBMITTED BY

MADHUMITHA S (113219041060)

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

ASSIGNMENT-04

Question:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than100cmssend “alert”to ibm cloudand display in devicerecentevents.

Uploaddocument with wokwi sharelink and images of ibmcloud.

Solution:

```
#include<WiFi.h>//libraryforwifi#include<Pu
bSubClient.h>//libraryforMQtt

#defineECHO_GPIO12
#defineTRIGGER_GPIO13
#defineMAX_DISTANCE_CM100//Maximumof5meters#include"Ultra
sonic.h"

Ultrasonicultrasonic(13,12);intd
istance;

voidcallback(char*subscribetopic,byte*payload,unsignedintpayloadLength);

//-------credentialsofIBMAccounts------

#defineORG"dv1snq"//IBMORGANITIONID
#defineDEVICE_TYPE"ESP32"//DevicetypementionedinibmwatsonIOTPlatform#define
DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT
Platform#defineTOKEN"45682367915"    //Token
Stringdata3;f
loath,t;


//--------Customisetheabovevalues--------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
Namechar publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type
ofeventperform andformatinwhich datato besend
charsubscribetopic[]="iot-2/cmd/command/fmt/String";//cmdREPRESENT
commandtype ANDCOMMANDISTESTOFFORMAT STRING
```

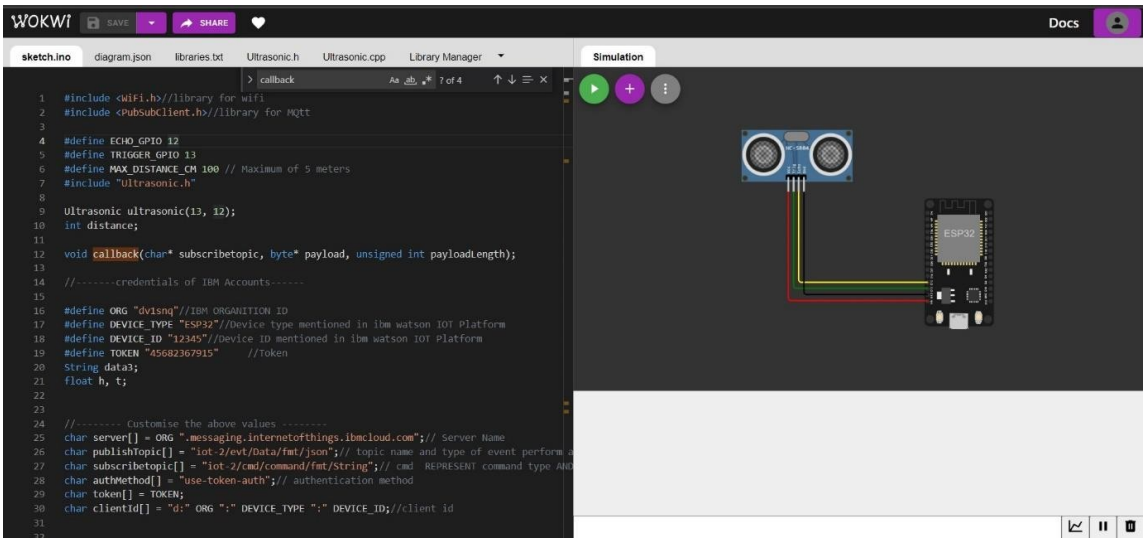
```
char authMethod[] = "use-token-auth";// authentication
methodchartoken[]=TOKEN;
charclientId[]="d:"ORG":"DEVICE_TYPE ":"DEVICE_ID;//clientid


//---------------------------------------
WiFiClient wifiClient; // creating the instance for
wificlientPubSubClient client(server, 1883, callback ,wifiClient);
//calling thepredefined client id by passing parameter like server
id,portandwificredential

voidsetup()//configureingtheESP32
{
  Serial.begin(115200);
  delay(10);Serial.prin
  tln();wificonnect();m
  qttconnect();
}

voidloop()//RecursiveFunction
{

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

  }

  delay(1000);


}

/*.....................................retrievingto
Cloud. ............................. */

void PublishData(float temp)
  {mqttconnect();//functioncallforconnectingtoibm
  /*
     creating theStringininformJSontoupdatethe datato ibmcloud
  */
  Stringpayload="{\"AlertDistance:\":";
```

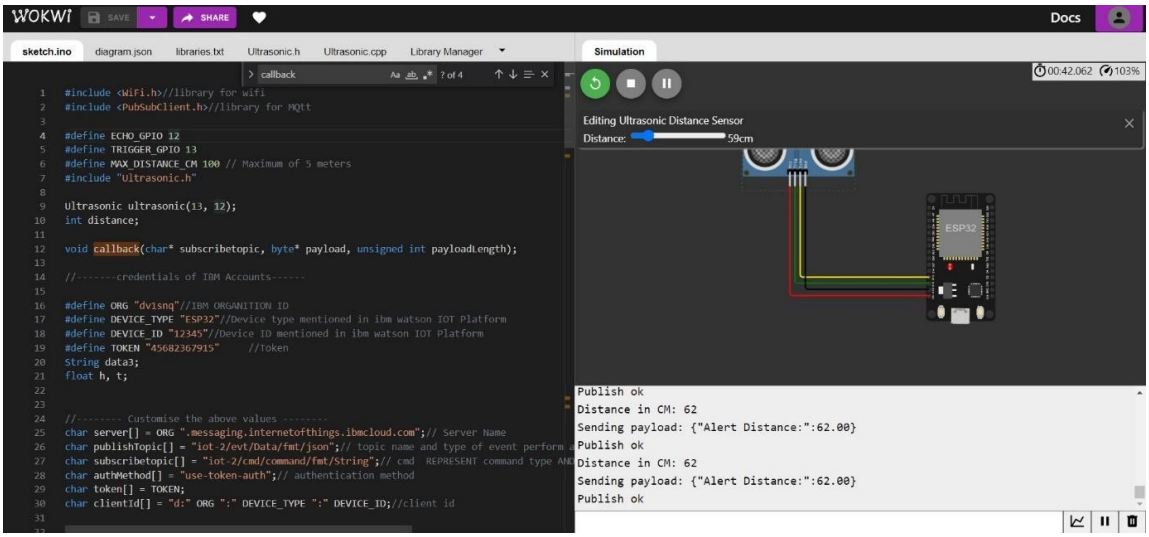
```
  payload +=
  temp;payload+="}
  ";


  Serial.print("Sendingpayload:");
  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("Publishfailed");
  }

}

voidmqttconnect(){
  if (!client.connected())
    {Serial.print("Reconnecting client to
    ");Serial.println(server);
    while(!!!client.connect(clientId,authMethod,token)){
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
voidwificonnect()//functiondefinationforwificonnect
{
  Serial.println();Serial.print("Co
  nnectingto");

  WiFi.begin("Wokwi-
GUEST","",6);//passingthewificredentialstoestablishtheconnection
  while (WiFi.status() != WL_CONNECTED)
    {delay(500);
    Serial.print(".");
  }
  Serial.println("");Serial.println
  ("WiFi
  connected");Serial.println("IP
  address:
  ");Serial.println(WiFi.localIP())
  ;
}
```

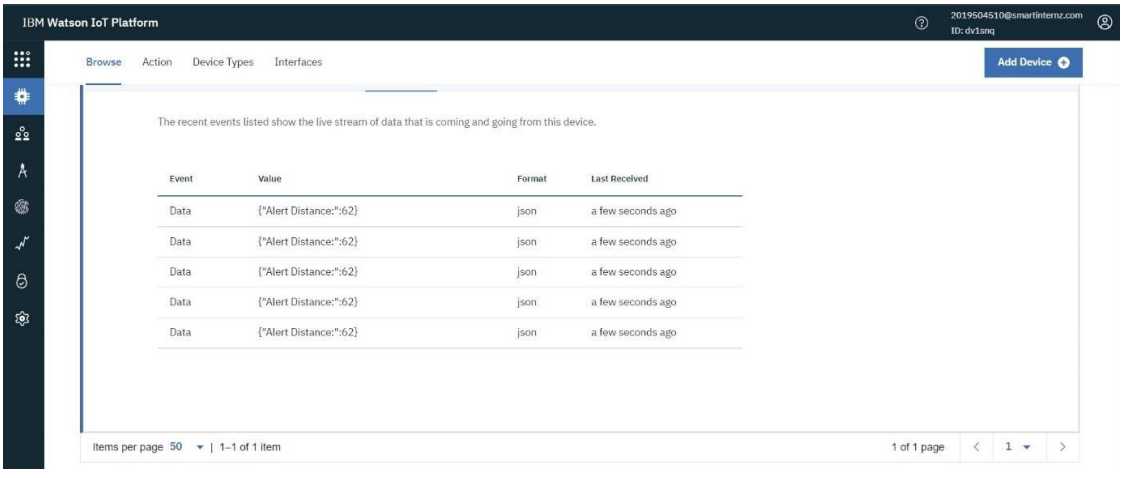
```
voidinitManagedDevice(){
  if (client.subscribe(subscribetopic))
    {Serial.println((subscribetopic));Serial.println("sub
    scribetocmdOK");
  }else{
    Serial.println("subscribetocmdFAILED");
  }
}

voidcallback(char*subscribetopic,byte*payload,unsignedintpayloadLength)
{

  Serial.print("callbackinvokedfortopic:");
  Serial.println(subscribetopic);
  for(inti=0;i<payloadLength;i++){
    //Serial.print((char)payload[i]);data
    3+= (char)payload[i];
  }
  Serial.println("data:"+data3);if
  (data3=="lighton")
  {
Serial.println(data3);
  }
  else
  {
Serial.println(data3);
  }
data3="";
}
```

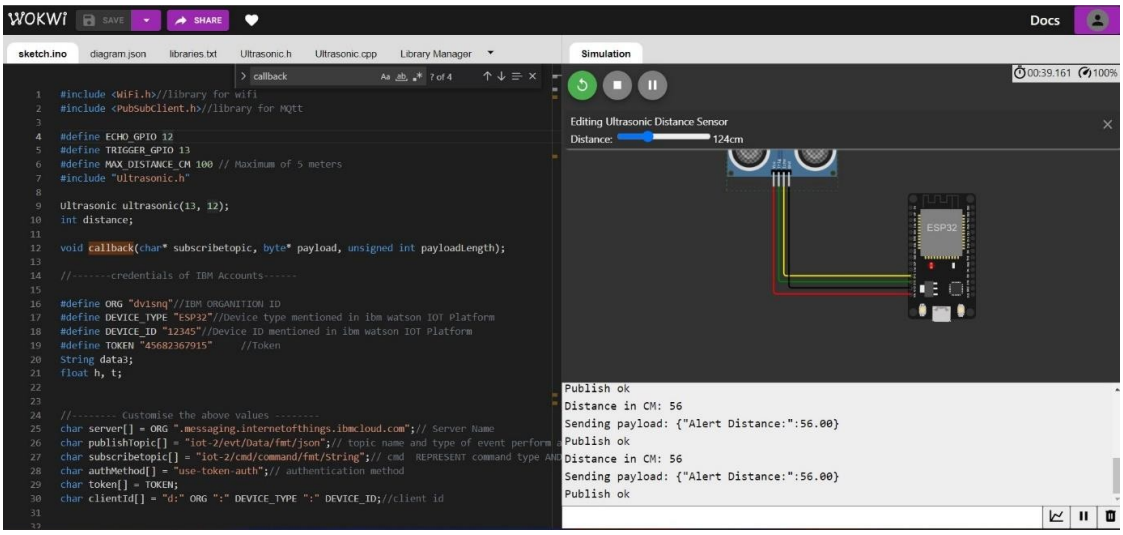
Wokwi

link:https://wokwi.com/projects/346659959540286034


WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Ultrasonic.h
Ultrasonic.cpp
Library Manager
Simulation
callback
? of 4
#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 "dv1snq"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "45682367915" //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
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
ESP32


WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Ultrasonic.h
Ultrasonic.cpp
Library Manager
Simulation
00:42.062
103%
Editing Ultrasonic Distance Sensor
Distance:
59cm
callback
? of 4
#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 "dv1snq"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "45682367915" //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
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
Publish ok
Distance in CM: 62
Sending payload: {"Alert Distance:":62.00}
Publish ok
Distance in CM: 62
Sending payload: {"Alert Distance:":62.00}
Publish ok
ESP32


IBM Watson IoT Platform
2019504510@smartinternz.com
ID: dv1snq
Browse
Action
Device Types
Interfaces
Add Device
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
Data {"Alert Distance:":62} json a few seconds ago
Data {"Alert Distance:":62} json a few seconds ago
Data {"Alert Distance:":62} json a few seconds ago
Data {"Alert Distance:":62} json a few seconds ago
Data {"Alert Distance:":62} json a few seconds ago
Items per page 50
1–1 of 1 item
1 of 1 page
1


WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Ultrasonic.h
Ultrasonic.cpp
Library Manager
Simulation
callback
7 of 4
00:39.161
100%
Editing Ultrasonic Distance Sensor
Distance:
124cm
ESP32
#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 "dv1snq"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "45682367915" //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
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
Publish ok
Distance in CM: 56
Sending payload: {"Alert Distance:":56.00}
Publish ok
Distance in CM: 56
Sending payload: {"Alert Distance:":56.00}
Publish ok