

Assignment-4
DistanceDetectionUsingUltrasonicSensor

AssignmentDate	19October2022
StudentName	Mr. Pragadeeshwaran.T
StudentRollNumber	111919106047
MaximumMarks	2 Marks

Question-1:

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

WOKWILINK: <https://wokwi.com/projects/345964118720643668>

CODE:

```
#include<WiFi.h>//library
forwifi#include<PubSubClient.h>//libraryfor
MQtt

voidcallback(char*subscribetopic,byte*payload,unsignedintpayloadL
ength);

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

#defineORG"f59trs"//IBMORGANITIONID
#defineDEVICE_TYPE"ultrasonicsensor"//Devicetypementionedinibmwat
sonIOTPlatform
#defineDEVICE_ID"distancedetection"//DeviceIDmentionedinibmwatson
IOTPlatform
#defineTOKEN"AlGMGaaF01nawa1QA3"
//TokenStringdata3;
floatdist;

//-----Customisetheabovevalues-----
char server[] = ORG
".messaging.internetofthings.ibmcloud.com";//ServerName
charpublishTopic[]="iot-2/evt/Data/fmt/json";//topicnameandtypeof
event perform andformatinwhich datatobesend
charsubscribetopic[]="iot-2/cmd/test/fmt/String";//
cmdREPRESENTcommandtypeANDCOMMANDIS TESTOFFORMATSTRING
charauthMethod[]="use-token-auth";//authenticationmethodchartoke
n[]=TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;//clientId

//
-
WiFiClientwifiClient;//creating theinstanceforwifiClient
```

```

PubSubClient client(server, 1883, callback, wifiClient);
//calling the predefined client id by passing parameter
likeserverid, port and wifi credential

int LED =
4; int trig =
5; int echo =
18; void setup()
{
  Serial.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED,
  OUTPUT); delay(10); wifi
  connect(); mqttconnect
  ();
}
void loop() //Recursive Function
{

  digitalWrite(trig, LOW); di
  gitalWrite(trig, HIGH); de
  layMicroseconds(10); digi
  talWrite(trig, LOW);
  float dur =
  pulseIn(echo, HIGH); float dist =
  (dur * 0.0343)/2; Serial.print
  ("Distance in
  cm"); Serial.println(dist);

  PublishData(dist);
  delay(1000);
  if (!client.loop())
    {mqttconnect();
    }
}

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

void PublishData(float dist)
{mqttconnect(); //function call for connecting to IBM
/*
  creating the String in form JSON to update the data
to IBM cloud
*/
String object;

```

```

if(dist<100)
{
    digitalWrite(LED,HIGH);Serial.print
    println("object is
    near");object="Near";
}
else
{
    digitalWrite(LED,LOW);Serial.print
    ln("no object found");object="No";
}

String payload =
"{\"distance\":\"";payload+=dist;
payload += ","
 "\"object\":\"";payload+=object
;
payload+= "}\"";

Serial.print("Sendingpayload:");
Serial.println(payload);


if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("Publish ok");// if it sucessfully upload
dataon the cloud then it will print publish ok in Serial monitor
orelseitwill printpublishfailed
}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();
}
}

```

```

void wificonnect() // function definition for wificonnect
{
    Serial.println(); Serial.print("
    Connecting to");

    WiFi.begin("Wokwi-GUEST", "", 6); // passing the wifi
    credentialsto 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("subscr
        ibetocmdOK");
    }
    else{
        Serial.println("subscribetocmdFAILED");
    }
}

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 == "Near")
    // {
    // Serial.println(data3);
    // digitalWrite(LED, HIGH);

    // }

    // else
    // {
    // Serial.println(data3);

```

```

//digitalWrite(LED,LOW);

//
}data3=""
;

}

```

OUTPUT:

When object is not near to the ultrasonic sensor

sketch.ino

diagram.json

libraries.txt

Library Manager

```

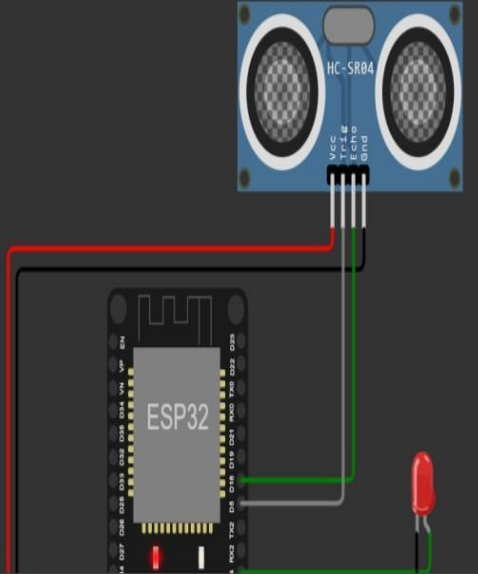
1  #include <WiFi.h>//library for wifi
2  #include <PubSubClient.h>//library for MQTT
3
4
5  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
6
7  //-----credentials of IBM Accounts-----
8
9  #define ORG "f59trs"//IBM ORGANITION ID
10 #define DEVICE_TYPE "ultrasonicsensor"//Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "distancedetection"//Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "ALGMGaaF01naw1QA3" //Token
13 String data3;
14 float dist;
15
16
17 //----- Customise the above values -----
18 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
19 char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and
20 char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type AND COMM
21 char authMethod[] = "use-token-auth";// authentication method
22 char token[] = TOKEN;
23 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
24
25
26 //-----
27 WiFiClient wificlient; // creating the instance for wificlient
28 PubSubClient client(server, 1883, callback ,wificlient); //calling the predefined client id
29
30 int LED = 4;
31 int trig = 5;
32 int echo = 18;
33 void setup()

```

Simulation

00:05.682

99%



no object found

Sending payload: {"distance":403.45,"object":"No"}

Publish ok

Distance in cm 233.00

no object found

Sending payload: {"distance":233.00,"object":"No"}

Publish ok

DatasenttotheIBMclouddevice whentheobjectis far

Browse Action Device Types Interfaces
Add Device +

▼ ■ distancedetection ● Connected ultrasonicsensor Device Oct 19, 2022 11:56 AM → ...

Identity Device Information Recent Events State Logs ✕

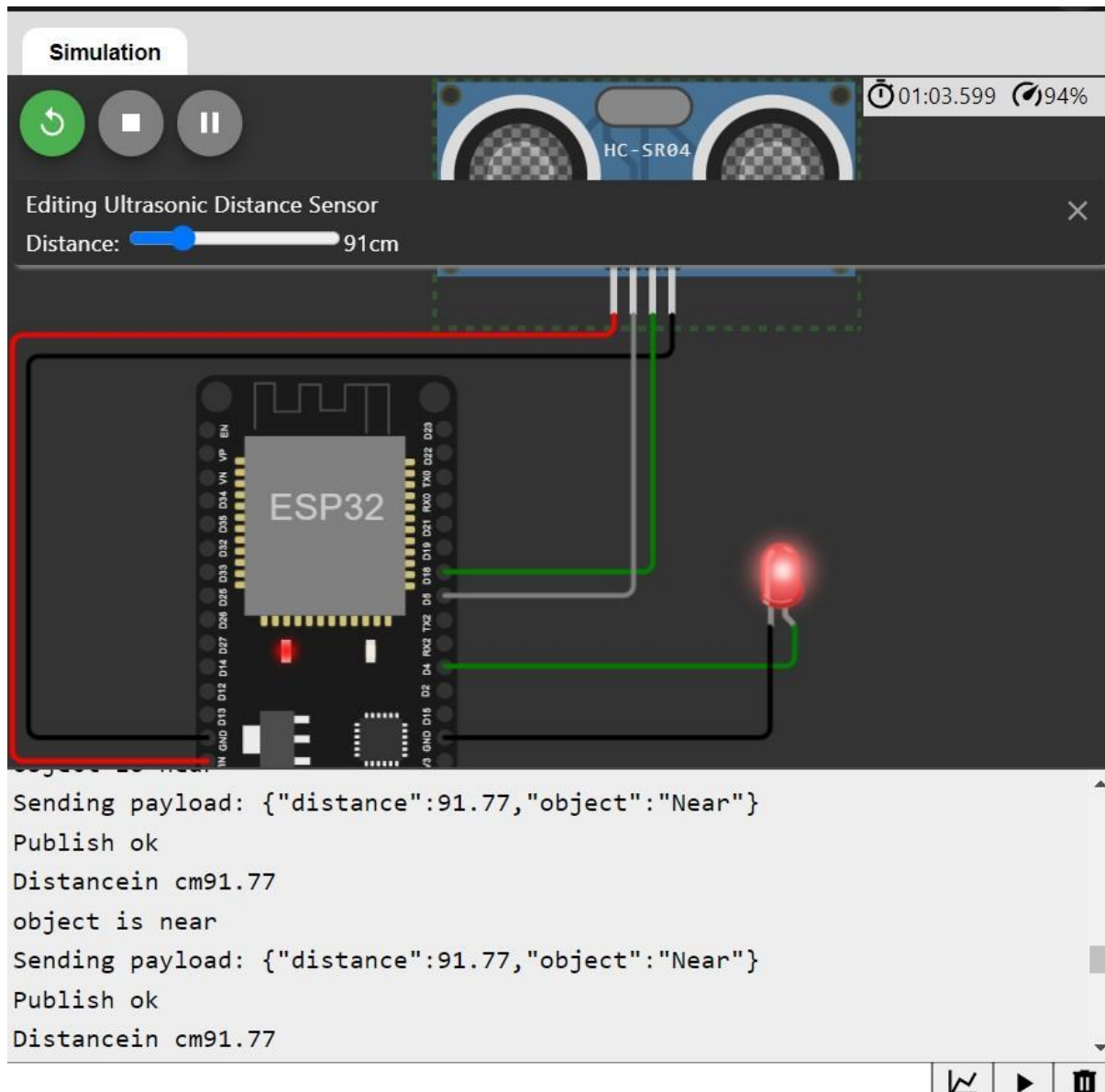
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago

Items per page 50 ▼ | 1–1 of 1 item

0 Simulations running

When object is near to the ultrasonic sensor



DatasenttotheIBMclouddevice whentheobjectisnear

Browser Action Device Types Interfaces

Q

Add Device +

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
distancedetection	Connected	ultrasonicsensor	Device	Oct 19, 2022 11:56 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":91.77,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.75,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.77,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.79,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.8,"object":"Near"}	json	a few seconds ago

0 Simulations running

<https://wokwi.com/projects/345964118720643668>

