ASSIGNMENT-4
DISTANCE DETECTION USING ULTRASONIC SENSOR

Date	26 October 2022
Team ID	PNT2022TMID17661
Name	SANTHOSH P
Student Roll Number	713319EC097
Maximum Marks	2 Marks

Question 1:

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

CODE:

```
#include <WiFi.h>//library for wifi
#include <Pubsubclient.h>//library for Hott
void callback(char subscribetopic, byte payload, unsigned int payloadLength);


#define ORG 4h0jp"//IBM ORGANITION IO
#define DEVICE_TYPE "ULTRASON
#define DEVICE_ID "DISTANCEDETECT
#define TOKEN "wuo5s7PR)ZSegvk&Rx"
String data3;
float dist;
char server[] -ORG ".messaging internetofthings.ibmcloud.com";// Server Name
char publishTopic[] "iot-2/evata/fmt/json";
char authMethod[]"use-token-auth";// authentication method
char token[] TOKEN;
char clientId[]"d:" ORG ":" DEVICE_TYPE":"DEVICE_ID;//client id

int LED = 4;
int trig 5;
int echo= 18;
void setic()
{
Serial.begin(115200);
}
```

esp32-blink.ino ● diagram.json ● libraries.txt ● Library Manager

```
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
pinMode(LED, OUTPUT);
delay(10);
wificonnect();
mqttconnect();
}
void loop()// Recursive Function
{

 digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  float dur = pulseIn(echo,HIGH);
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distancein 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 in form JSon to update the data to ibm cloud
```

```
    creating the String in in form JSon to update the data to ibm cloud
  */
  String object;
  if (dist <100)
  {
    digitalWrite(LED,HIGH);
    Serial.println("object is near");
    object = "Near";
  }
  else
  {
    digitalWrite(LED,LOW);
    Serial.println("no object found");
    object = "No";
  }

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


  Serial.print("Sending payload: ");
  Serial.println(payload);
```

esp32-blink.ino ● diagram.json ● libraries.txt ● Library Manager

```
  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());
```

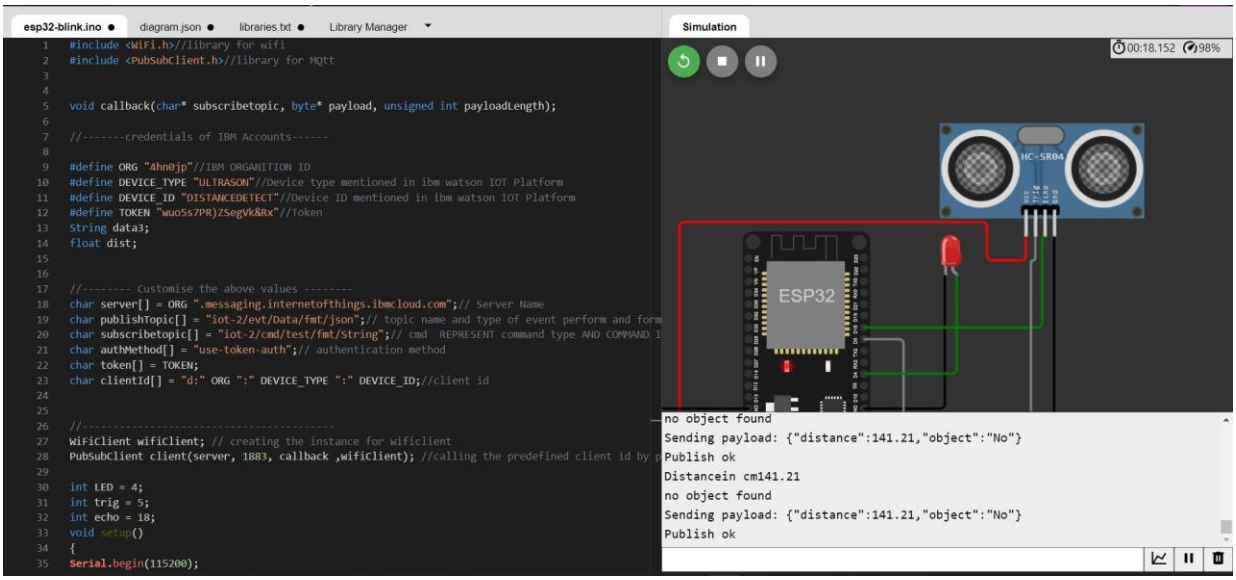
esp32-blink.ino ● diagram.json ● libraries.txt ● Library Manager

```
  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=="Near")
//   {
// Serial.println(data3);
```

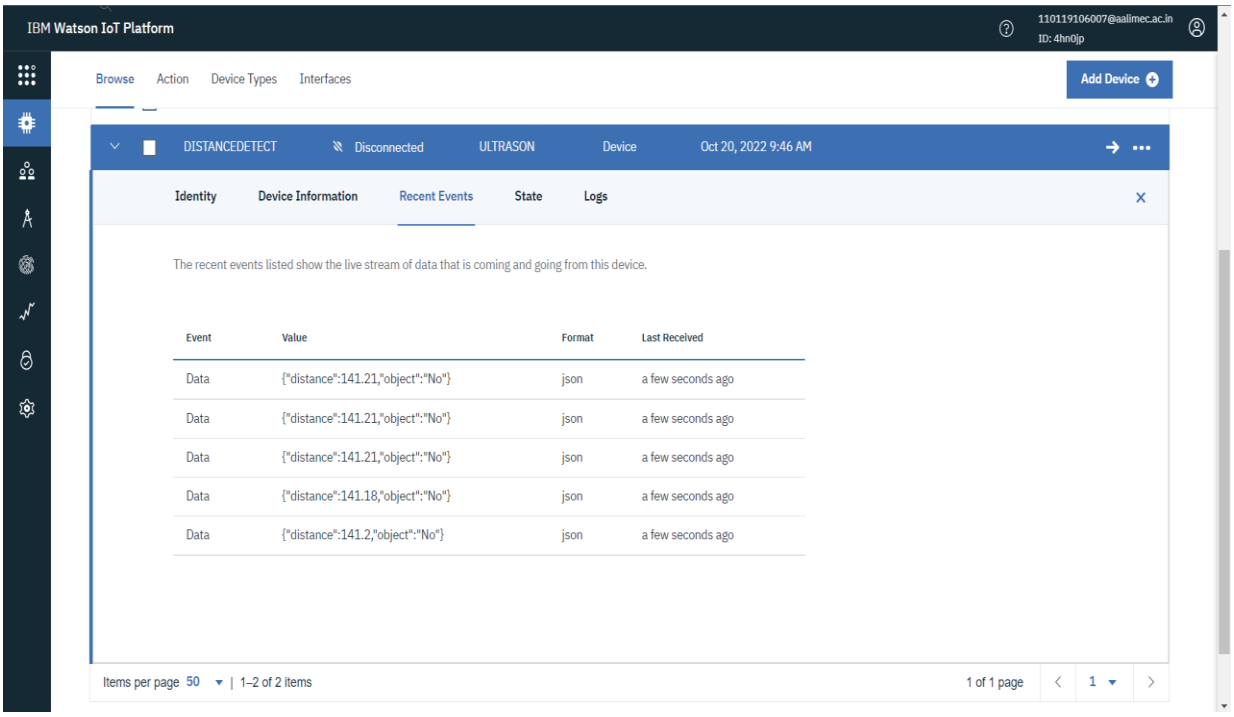
esp32-blink.ino ● diagram.json ● libraries.txt ● Library Manager

```
142  }
143
144  void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
145  {
146
147    Serial.print("callback invoked for topic: ");
148    Serial.println(subscribetopic);
149    for (int i = 0; i < payloadLength; i++) {
150      //Serial.print((char)payload[i]);
151      data3 += (char)payload[i];
152    }
153
154  //   Serial.println("data: "+ data3);
155  //   if(data3=="Near")
156  //   {
157  // Serial.println(data3);
158  // digitalWrite(LED,HIGH);
159
160  //   }
161
162  //   else
163  //   {
164  // Serial.println(data3);
165  // digitalWrite(LED,LOW);
166
167  //   }
168  data3="";
169
170
171  }
```

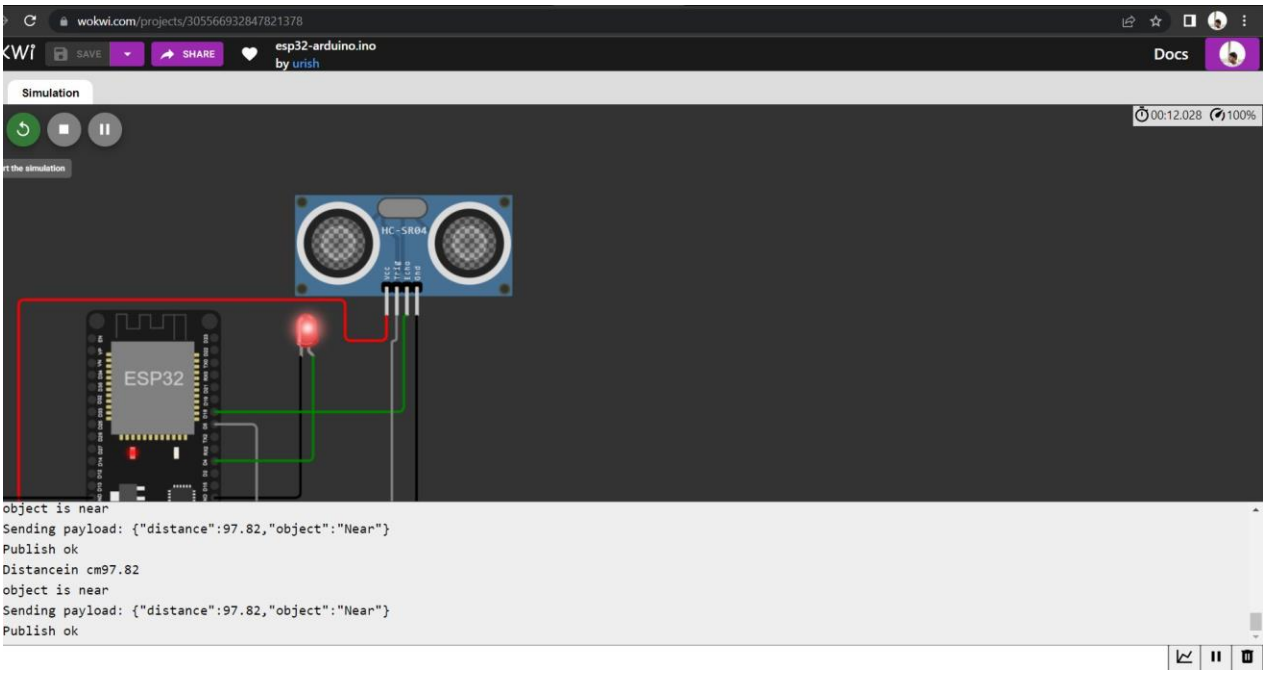
OUTPUT:



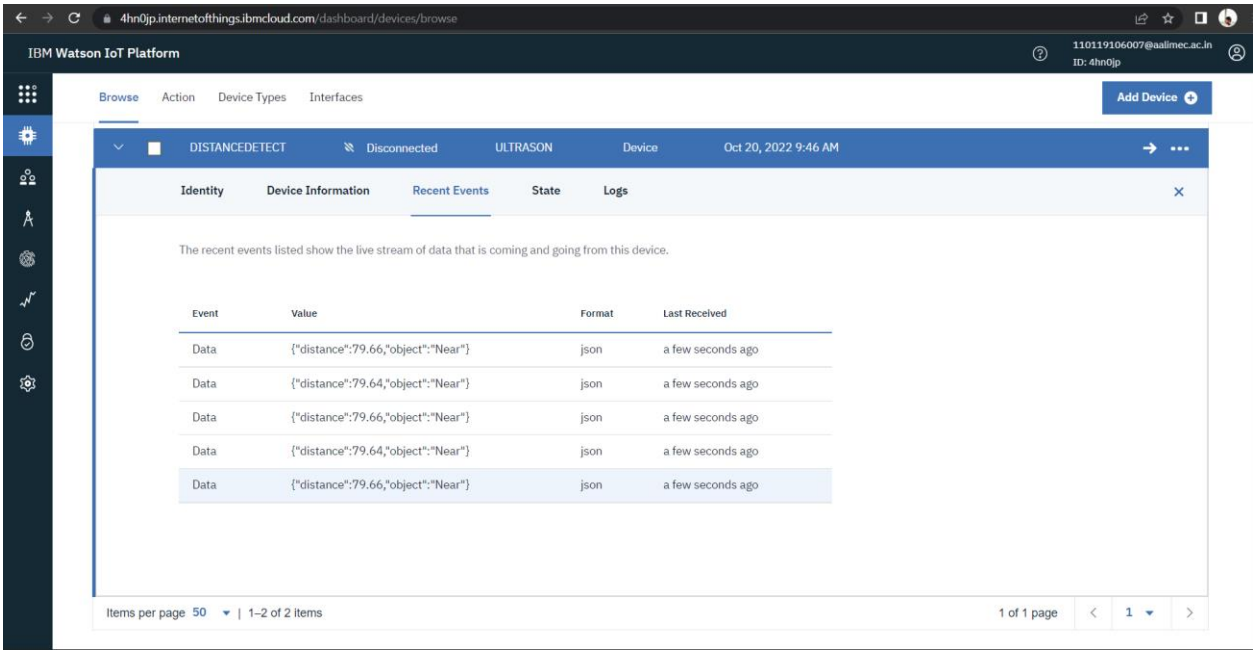
Data send to the IBM cloud device when the object is far



when object is near to the ultrasonic sensor



Data sent to the IBM Cloud Device when the object is near



https://wokwi.com/projects/305566932847821378