ASSIGNMENT-4

DISTANCE DETECTION USING ULTRASONIC SENSOR

TEAM ID: PNT2022TMID52298
NAME: ANUSHA K
REGISTER NUMBER: 963519106009
PROJECT TITLE: Gas leakage monitering and alerting system.

Question1 :

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

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 "4hn0jp"//IBM ORGANITION ID
#define DEVICE_TYPE "ULTRASON"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "DISTANCEDETECT"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "wuo5s7PR)ZSegVk&Rx"//Token
String data3;
float dist;


//-------- 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

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

esp32 blink.ino | diagram.json

```
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
pinMode(LED, OUTPUT);
delay(10);
wificonnect();
```

```
}
void loop()// Recursive Function
{
mqttconnect();
}
  digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
        dur =    .     (ec ho,     );
    al dT st = (du r * 6 . 0343) /2;
  Serial.print ("Distancein cm");
  Serial.println(dist);
```

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

```
    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

```
98
99    if (client.publish(publishTopic, (char*) payload.c_str())) {
100     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
101   } else {
102     Serial.println("Publish failed");
103   }
104
105 }
106 void mqttconnect() {
107   if (!client.connected()) {
108     Serial.print("Reconnecting client to ");
109     Serial.println(server);
110     while (!!client.connect(clientId, authMethod, token)) {
111       Serial.print(".");
112       delay(500);
113     }
114
115     initManagedDevice();
116     Serial.println();
117   }
118 }
119 void wificonnect() //function defination for wificonnect
120 {
121   Serial.println();
122   Serial.print("Connecting to ");
123
124   WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
125   while (WiFi.status() != WL_CONNECTED) {
126     delay(500);
127     Serial.print(".");
128   }
129   Serial.println("");
130   Serial.println("WiFi connected");
131   Serial.println("IP address: ");
132   Serial.println(WiFi.localIP());
```

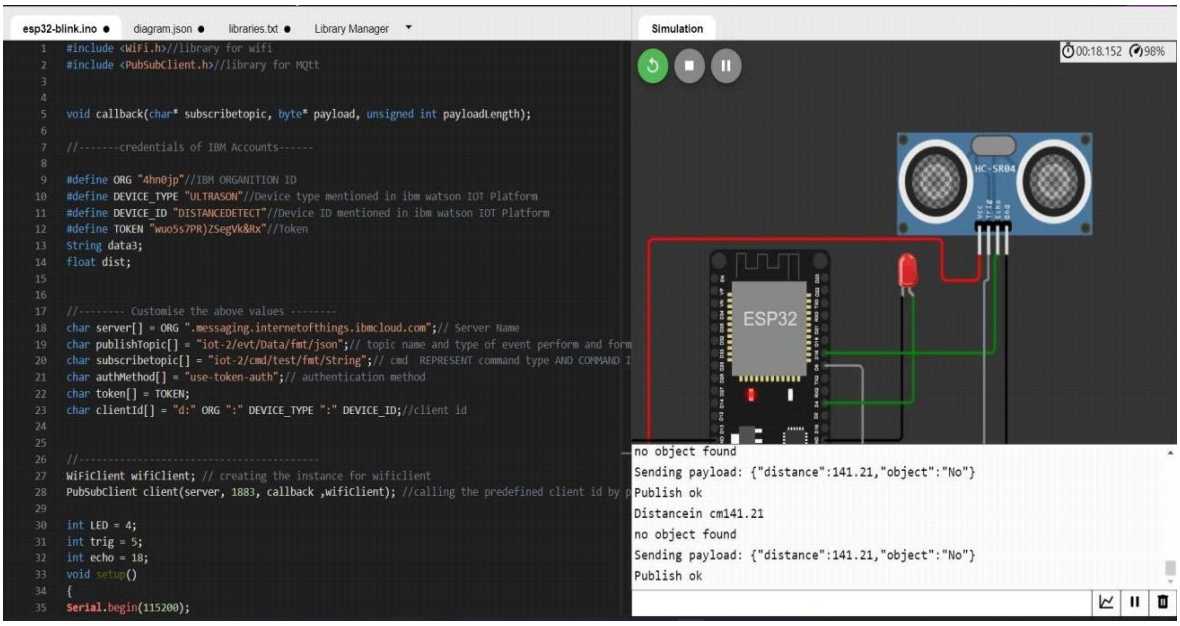
```
            initManagedDevice() {
 .<           lievi uA ribe(sâsribeioQ )  (
 .  "        SeriaZ.rint T_( (subscr Tbet opi c));
             SeriaZ ¿r i nt T n(" s ubs r r Tbe to c od OK ");

                        (                        );
141     }
142   }
```

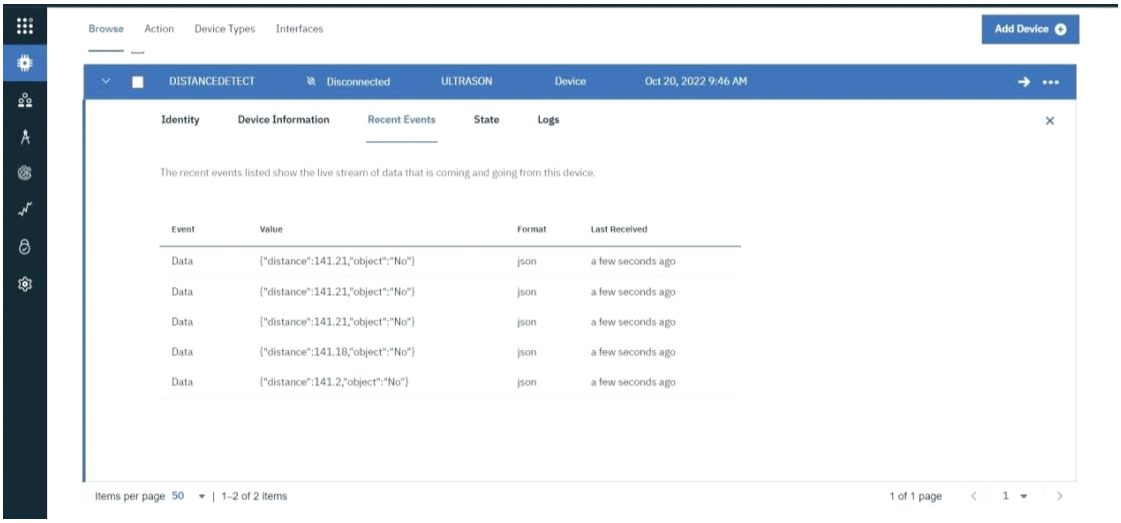
esp32-blink.ino ● diagram.json ● libraries.txt ● Library Manager

```
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:

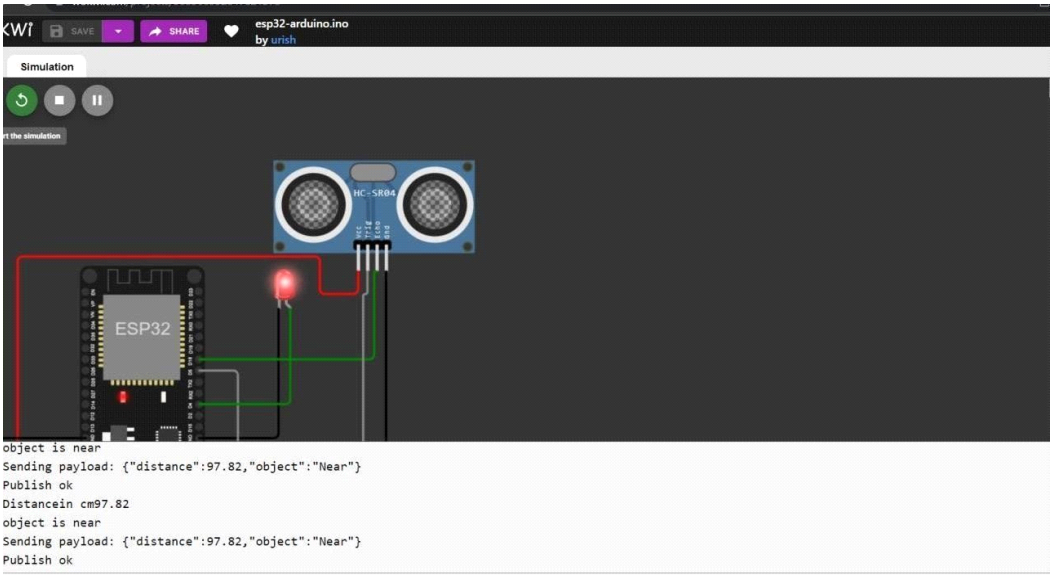


Data send to the IBM cloud device when the object is far

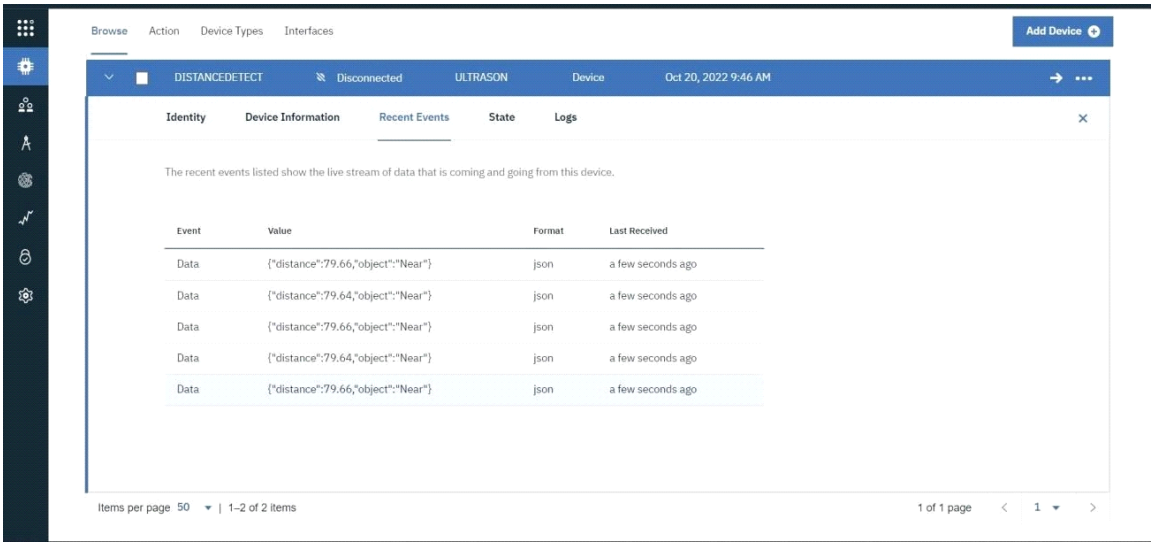


when object is near to the ultrasonic sensor

when object is near to the ultrasonic sensor



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


Browse
Action
Device Types
Interfaces
Add Device
DISTANCEDETECT
Disconnected
ULTRASON
Device
Oct 20, 2022 9:46 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":79.66,"object":"Near"}
json
a few seconds ago
Data
{"distance":79.64,"object":"Near"}
json
a few seconds ago
Data
{"distance":79.66,"object":"Near"}
json
a few seconds ago
Data
{"distance":79.64,"object":"Near"}
json
a few seconds ago
Data
{"distance":79.66,"object":"Near"}
json
a few seconds ago
Items per page 50
1–2 of 2 items
1 of 1 page
1