Assignment 4

WOKWI STIMULATOR

QUESTION:

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

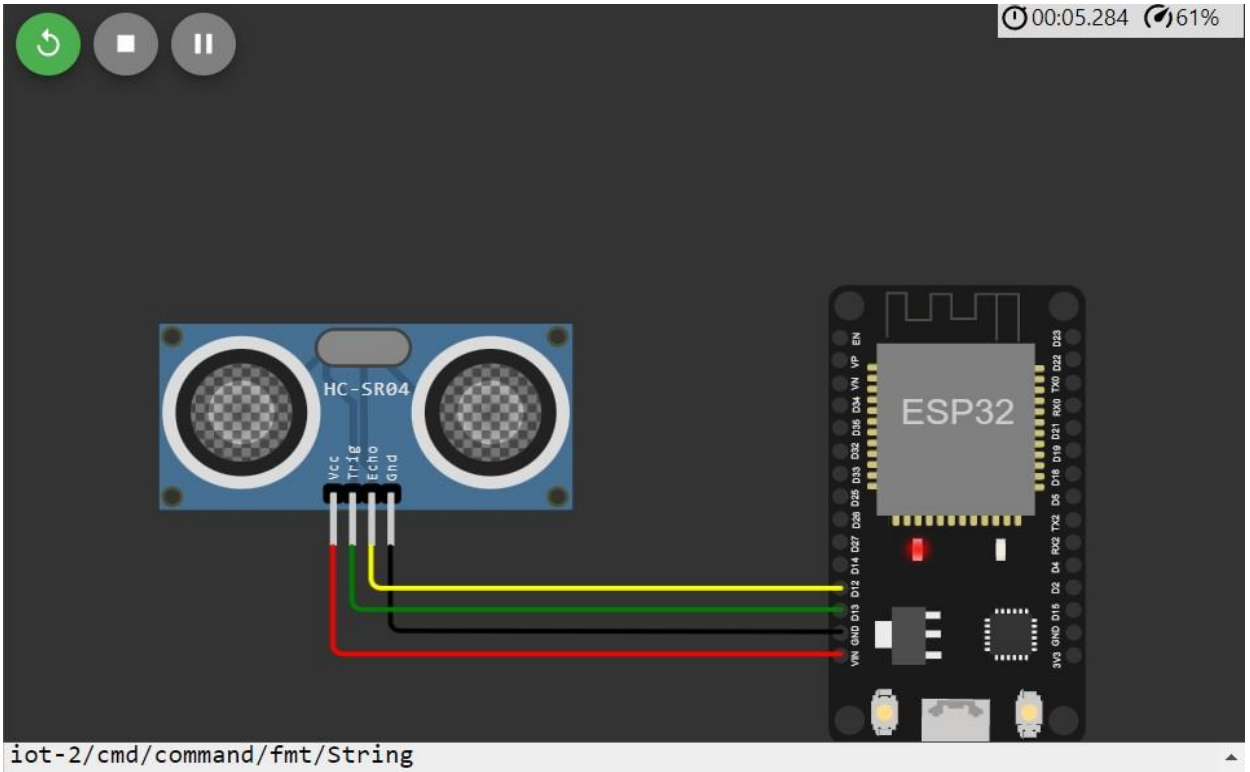
CODE:

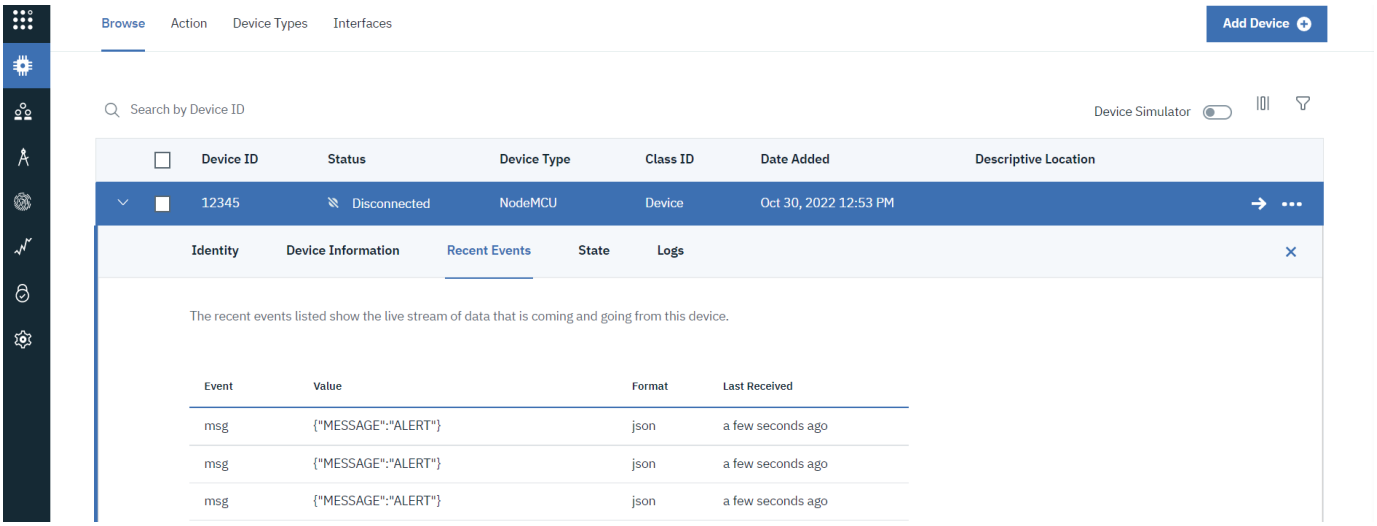
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define TrigPIN 15
#define EchoPIN 4
#define MINDIST 100



void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "7c6kj2"
#define DEVICE_TYPE "Nodemcu"
#define DEVICE_ID "2345"
#define TOKEN "06072002"   //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/msg/fmt/json";// topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/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


void setup()// configureing the ESP32
{
  Serial.begin(115200);
```

```
pinMode(TrigPIN, OUTPUT); digitalWrite(TrigPIN, LOW);pinMode(EchoPIN, INPUT);
delay(10); Serial.println(); wificonnect(); mqttconnect();
}

void loop()// Recursive Function
{
unsigned long t1;unsigned long t2;
unsigned long pulse_Width;float distance;

digitalWrite(TrigPIN, HIGH);delayMicroseconds(10); digitalWrite(TrigPIN, LOW);

pulse_Width = pulseIn(EchoPIN,HIGH);

distance= pulse_Width *0.034 / 2;

if(distance<100)
{
PublishData();
}

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



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

void PublishData() {
mqttconnect();//function call for connecting to ibm
/*
creating the String in in form JSon to update the data to ibm cloud
*/
String payload = "{\"MESSAGE\":\"ALERT\"}";


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 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());
  }

  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)
  {

  }
```

Output:


00:05.284
61%
HC-SR04
Vcc
Trig
Echo
Gnd
ESP32
iot-2/cmd/command/fmt/String
subscribe to cmd OK
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}


Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345
Disconnected
NodeMCU
Device
Oct 30, 2022 12:53 PM
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
msg
{"MESSAGE":"ALERT"}
json
a few seconds ago
msg
{"MESSAGE":"ALERT"}
json
a few seconds ago
msg
{"MESSAGE":"ALERT"}
json
a few seconds ago