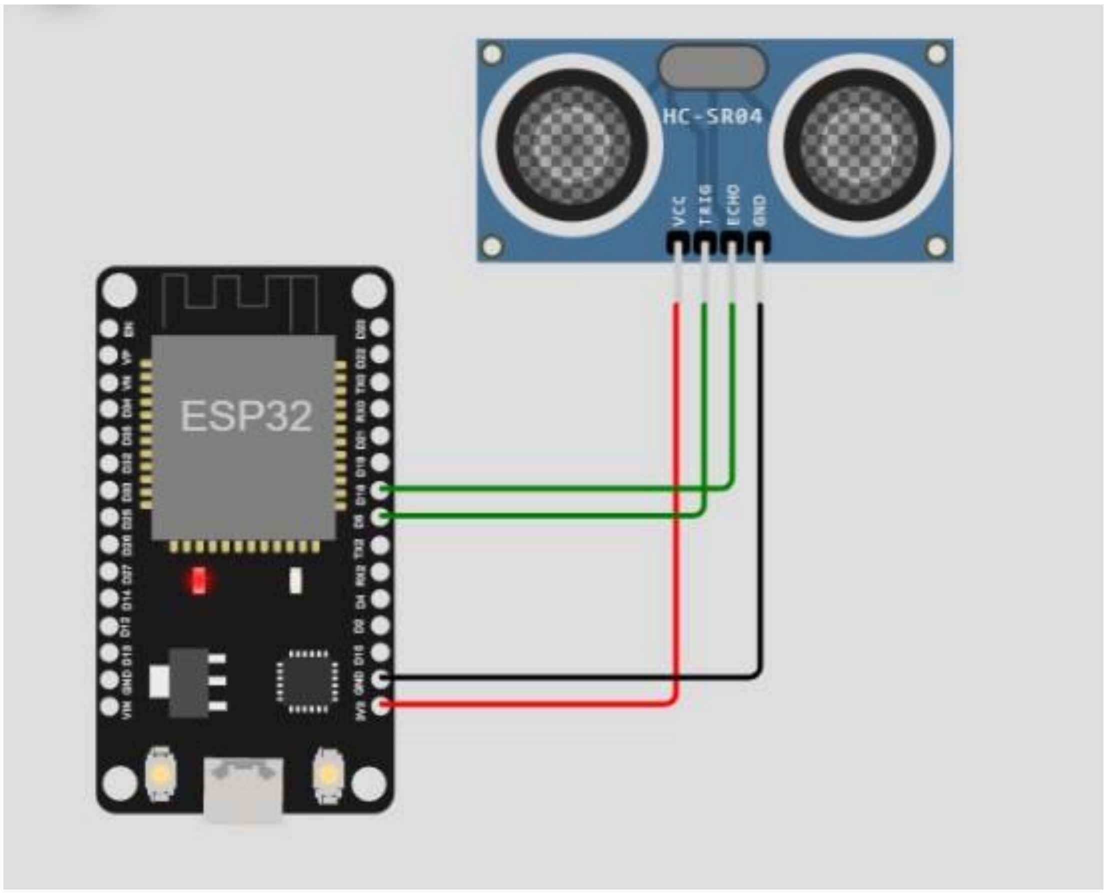
ASSIGNMENT - 3

Build wowki product, use ultrasonic sensor and detect the distance from the object. Whenever distance is less than 100cms upload the value to the ibm cloud.in recent device events upload the data from wokwi.

WOKWI share link: https://wokwi.com/projects/364898364098433025

Connections image:



WOKWI Code:

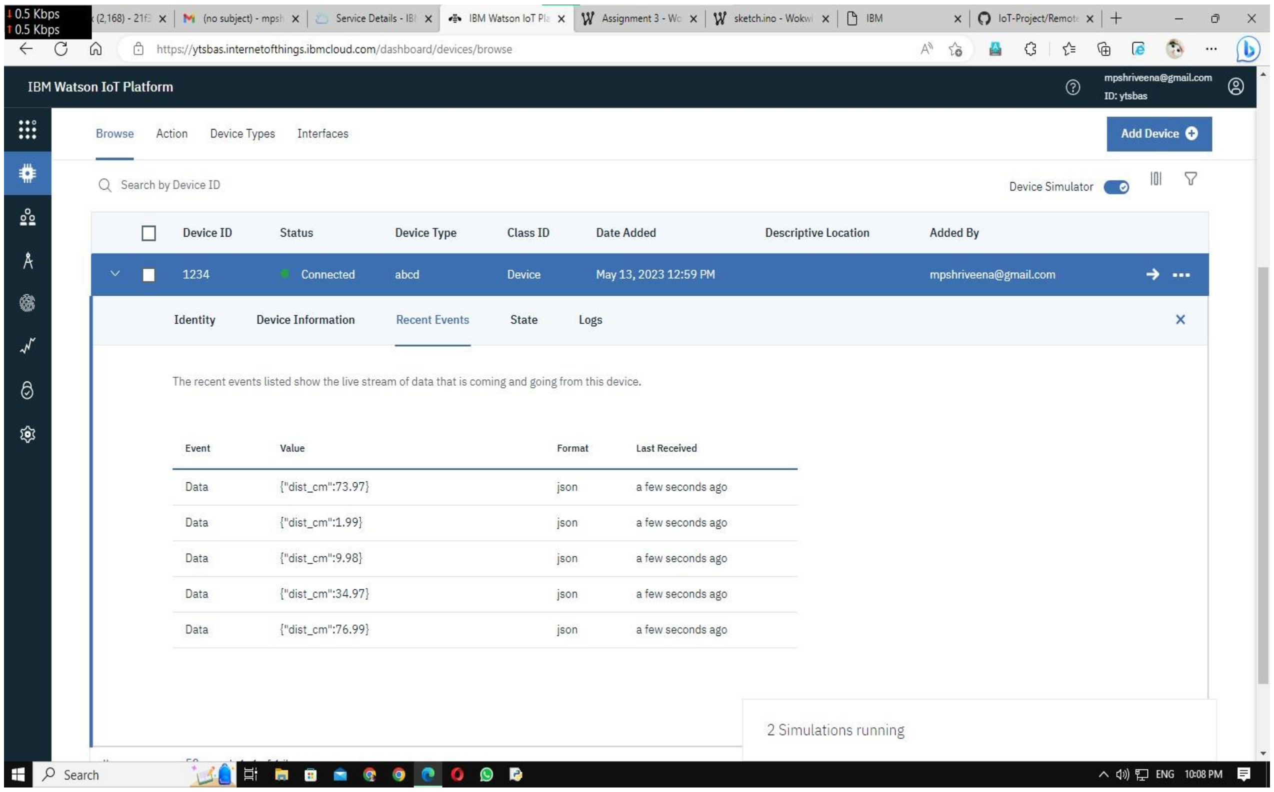
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

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

#define ORG "ytsbas"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
```

```
#define TOKEN "12345678"     //Token
String data3;
float dist_cm;


//-------- 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/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

int t_p=5;
int e_p=18;
float ss=0.034;
long durn;
void setup() {
Serial.begin(115200);
pinMode(t_p, OUTPUT);
pinMode(e_p, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(t_p, LOW);
delayMicroseconds(2);
digitalWrite(t_p,HIGH);
delayMicroseconds(10);
digitalWrite(t_p, LOW);
durn=pulseIn(e_p,HIGH);
dist_cm=durn*ss/2;
Serial.print("distance= ");
Serial.println(dist_cm);
delay(1000);
PublishData(dist_cm);
```

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

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

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


  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)
{

  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<"100")
  {
Serial.println(data3);
  }
  else
  {
Serial.println("No Data");
```

```
   }
data3="";
}
```

IBM Cloud recent events image:



IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
1234	Connected	abcd	Device	May 13, 2023 12:59 PM		mpshriveena@gmail.com

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	{"dist_cm":73.97}	json	a few seconds ago
Data	{"dist_cm":1.99}	json	a few seconds ago
Data	{"dist_cm":9.98}	json	a few seconds ago
Data	{"dist_cm":34.97}	json	a few seconds ago
Data	{"dist_cm":76.99}	json	a few seconds ago

2 Simulations running