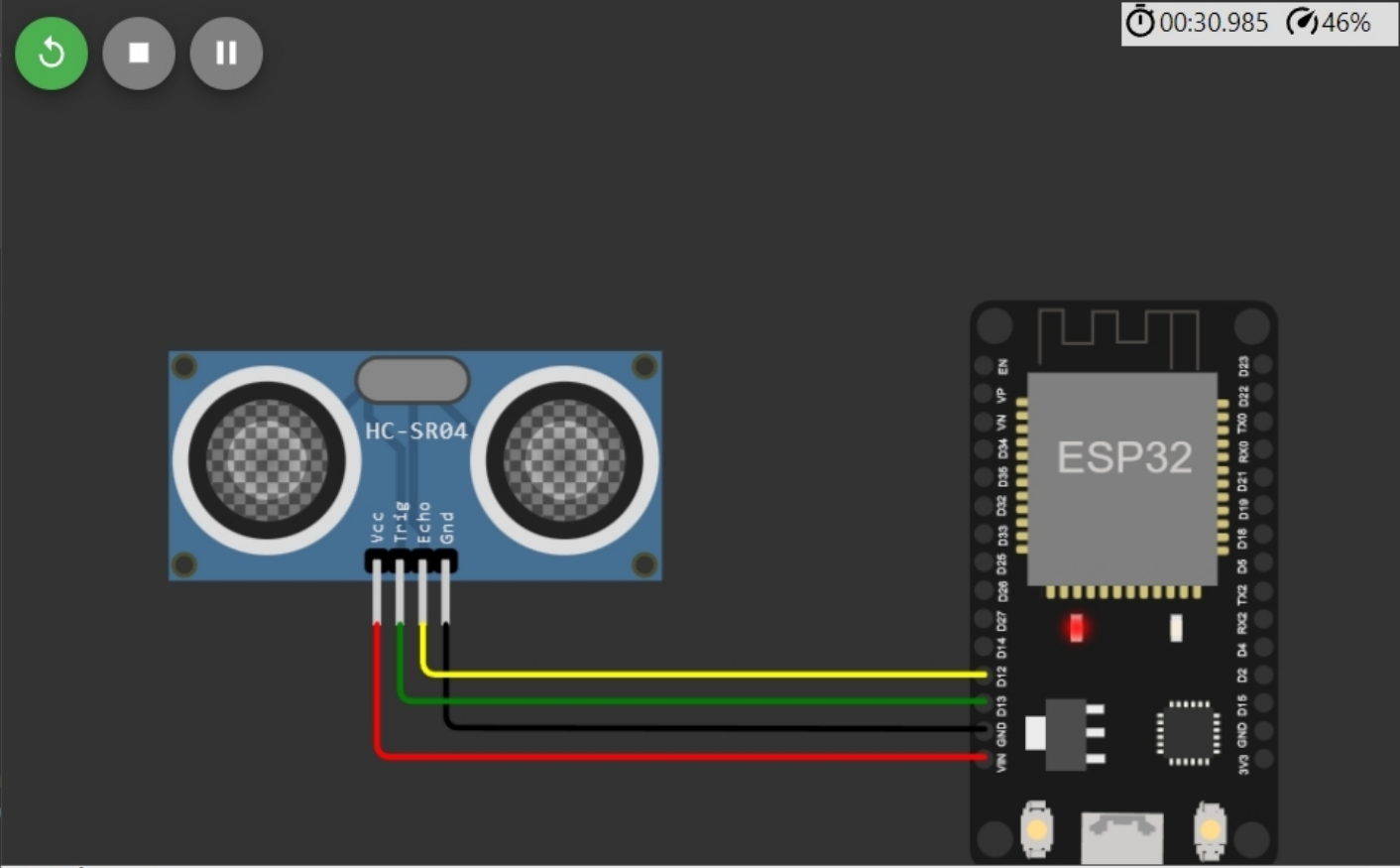


```

1 #include <WiFi.h>//library for wifi
2 #include <PubSubClient.h>//library for MQTT
3 #define TrigPIN 15
4 #define EchoPIN 4
5 #define MINDIST 100
6
7
8
9 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "bjjiwr"//IBM ORGANITION ID
14 #define DEVICE_TYPE "hinata"//Device type mentioned in ibm watson IOT Platform
15 #define DEVICE_ID "0812"//Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "12345678" //Token
17 String data3;
18 float h, t;
19
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform a
24 char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND
25 char authMethod[] = "use-token-auth";// authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
28
29
30 //-----
31 WiFiClient wificlient; // creating the instance for wificlient
32 PubSubClient client(server, 1883, callback ,wificlient); //calling the predefined client
33
34
35 void setup()// configureing the ESP32
36

```



Publish ok
 Sending payload: {"MESSAGE":"ALERT"}
 Publish ok
 Sending payload: {"MESSAGE":"ALERT"}
 Publish ok
 Sending payload: {"MESSAGE":"ALERT"}
 Publish ok

```
34
35 void setup()// configureing the ESP32
36 {
37   Serial.begin(115200);
38   pinMode(TrigPIN, OUTPUT);
39   digitalWrite(TrigPIN, LOW);
40   pinMode(EchoPIN, INPUT);
41   delay(10);
42   Serial.println();
43   wificonnect();
44   mqttconnect();
45 }
46
47 void loop()// Recursive Function
48 {
49   unsigned long t1;
50   unsigned long t2;
51   unsigned long pulse_width;
52   float distance;
53
54   digitalWrite(TrigPIN, HIGH);
55   delayMicroseconds(10);
56   digitalWrite(TrigPIN, LOW);
57
58   pulse_width = pulseIn(EchoPIN,HIGH);
59
60   distance= pulse_width *0.034 / 2;
61
62   if(distance<100)
63   {
64     PublishData();
65   }
66
67   delay(1000);
68   if (!client.loop()) {
69     mattconnect();
```

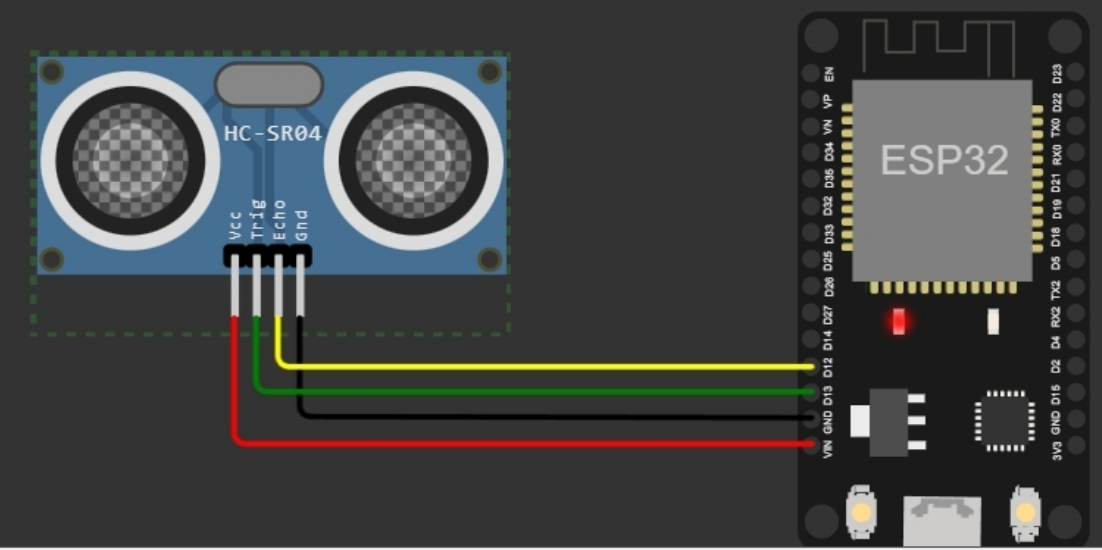
Simulation

01:02.193 81%

Reset Stop Pause

Editing Ultrasonic Distance Sensor

Distance: 228cm



Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok

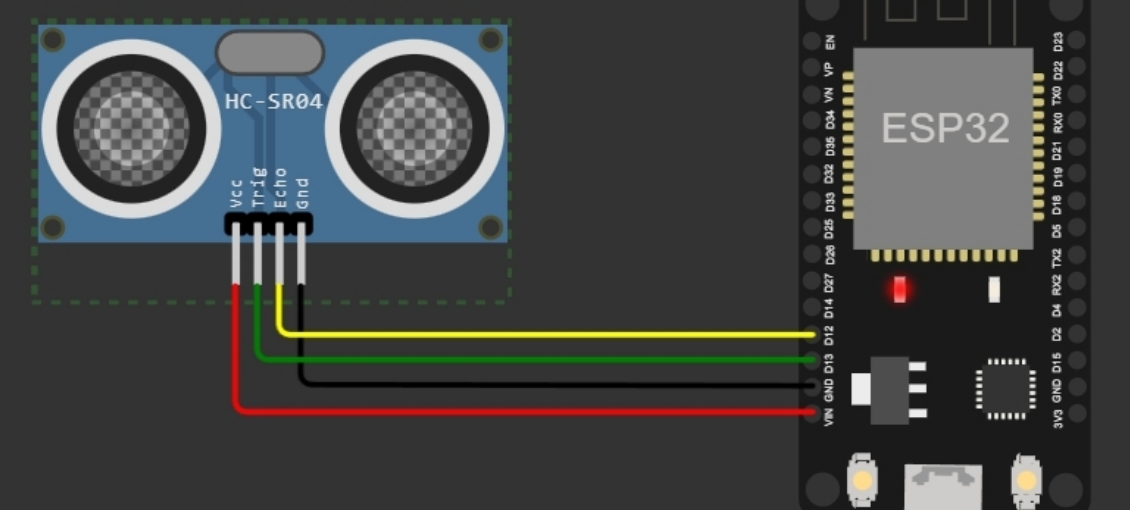
```
67 delay(1000);
68 if (!client.loop()) {
69   mqttconnect();
70 }
71 }
72
73
74
75 /*.....retrieving to Cloud.....
76
77 void PublishData() {
78   mqttconnect();//function call for connecting to ibm
79   /*
80   | creating the String in in form JSON to update the data to ibm cloud
81   */
82   String payload = "{\"MESSAGE\":\"ALERT\"}";
83
84
85   Serial.print("Sending payload: ");
86   Serial.println(payload);
87
88
89   if (client.publish(publishTopic, (char*) payload.c_str())) {
90     Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it wi
91   } else {
92     Serial.println("Publish failed");
93   }
94 }
95
96
97
98 void mqttconnect() {
99   if (!client.connected()) {
100     Serial.print("Reconnecting client to ");
101     Serial.println(server);
```

Simulation

01:06.743 83%

Editing Ultrasonic Distance Sensor

Distance: 228cm



Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok
Sending payload: {"MESSAGE":"ALERT"}
Publish ok

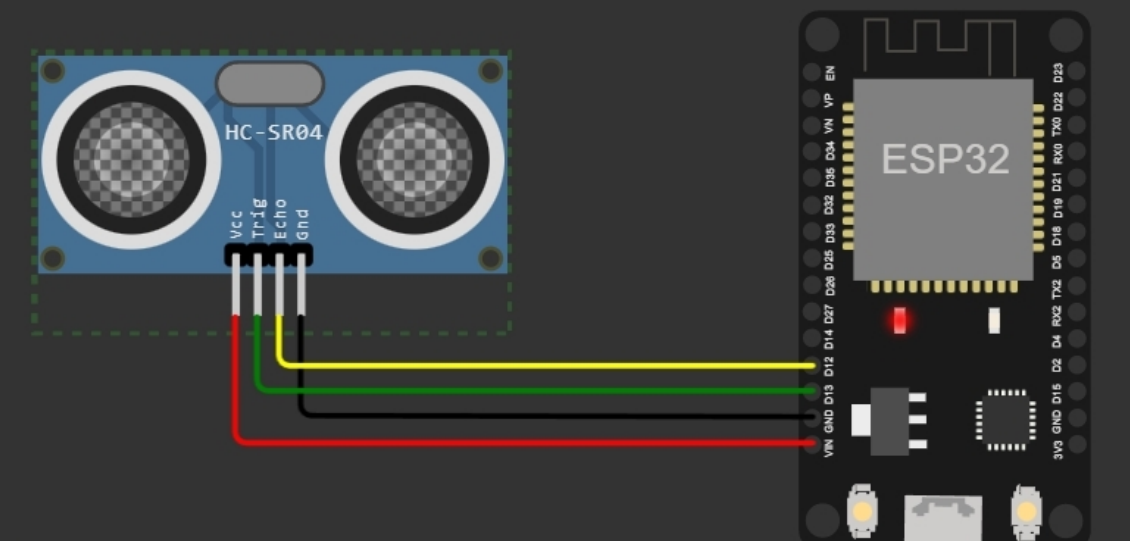

```
107   initManagedDevice();
108   Serial.println();
109 }
110 }
111 void wificonnect() //function defination for wificonnect
112 {
113   Serial.println();
114   Serial.print("Connecting to ");
115
116   WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connect
117   while (WiFi.status() != WL_CONNECTED) {
118     delay(500);
119     Serial.print(".");
120   }
121   Serial.println("");
122   Serial.println("WiFi connected");
123   Serial.println("IP address: ");
124   Serial.println(WiFi.localIP());
125 }
126
127 void initManagedDevice() {
128   if (client.subscribe(subscribetopic)) {
129     Serial.println((subscribetopic));
130     Serial.println("subscribe to cmd OK");
131   } else {
132     Serial.println("subscribe to cmd FAILED");
133   }
134 }
135
136 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
137 {
138
139 }
```

Simulation

01:12.778 93%

Editing Ultrasonic Distance Sensor

Distance: 228cm



Publish ok
Sending payload: {"MESSAGE": "ALERT"}
Publish ok
Sending payload: {"MESSAGE": "ALERT"}
Publish ok
Sending payload: {"MESSAGE": "ALERT"}
Publish ok

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

Event	Value	Format	Last Received
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago
Data	{"MESSAGE":"ALERT"}	json	a few seconds ago