

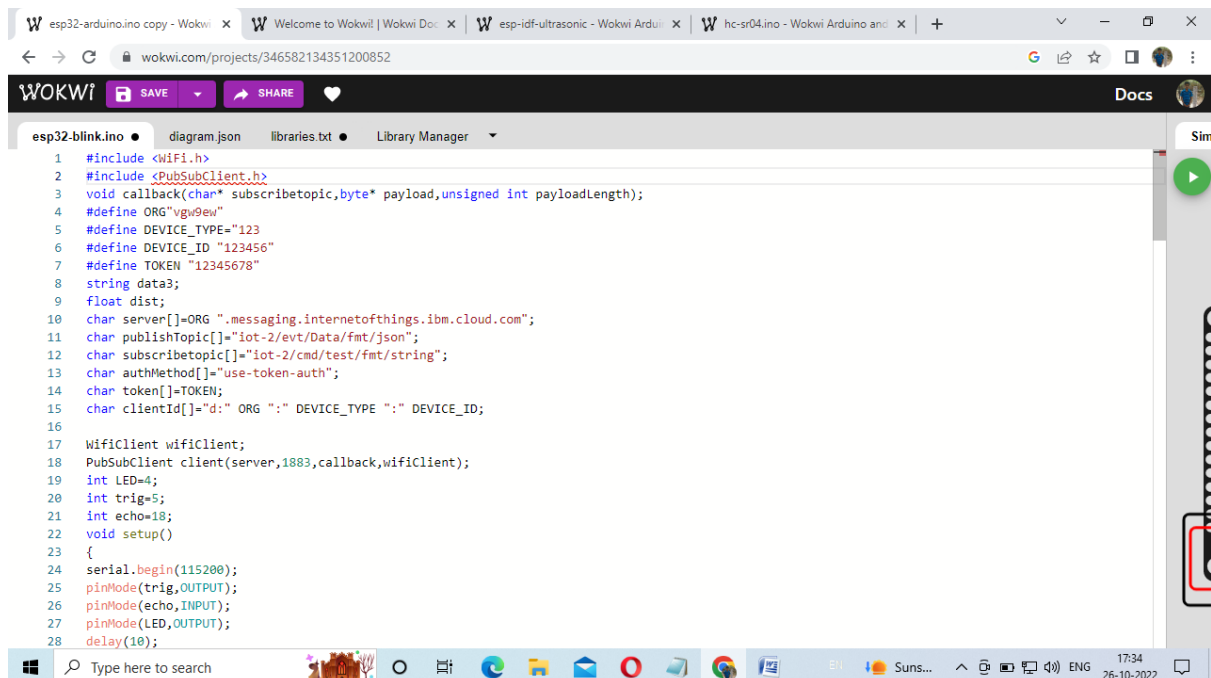
Assignment -4
Distance Detector using Ultrasonic Sensor

Assignment Date	26 September 2022
Student Name	S.KRITHIK GOKUL
Student Roll Number	811519106071
Maximum Marks	2 Marks

Question-1:

Write code and Connection in wokwi for ultrasonic sensor. Whenever distance is less than 100 Cms send "alert" to ibm cloud and display in device recent events.

Wokwi Link: <https://wokwi.com/projects/346582134351200852>



```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "vgu9ev"
5 #define DEVICE_TYPE "123"
6 #define DEVICE_ID "123456"
7 #define TOKEN "12345678"
8 string data3;
9 float dist;
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char subscribTopic[] = "iot-2/cmd/test/fmt/string";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16
17 WifiClient wifiClient;
18 PubSubClient client(server, 1883, callback, wifiClient);
19 int LED = 4;
20 int trig = 5;
21 int echo = 18;
22 void setup()
23 {
24   serial.begin(115200);
25   pinMode(trig, OUTPUT);
26   pinMode(echo, INPUT);
27   pinMode(LED, OUTPUT);
28   delay(10);
```

Wokwi IDE interface showing the code for `esp32-blink.ino`. The code is as follows:

```
27 pinMode(LED, OUTPUT);
28 delay(10);
29 wifiConnect();
30 mqttConnect();
31 }
32 void loop()
33 {
34   digitalWrite(trig, LOW);
35   digitalWrite(trig, HIGH);
36   delayMicroseconds(10);
37   digitalWrite(trig, LOW);
38   float dur=pulseIn(echo, HIGH);
39   float dist=(dur*0.0343)/2;
40   Serial.print("Distance in cm");
41   Serial.println(dist);
42   PublishData(dist);
43   delay(1000);
44   if(!client.loop()){
45     mqttConnect();
46   }
47 }
48 void PublishData(float dist)
49 {
50   mqttConnect();
51   String object;
52   if(dist<100)
53   {
54     digitalWrite(LED, HIGH);
```

Wokwi IDE interface showing the code for `esp32-blink.ino`. The code is as follows:

```
54 digitalWrite(LED, HIGH);
55 Serial.println("object is near");
56 object="Near";
57 }
58 else
59 {
60   digitalWrite(LED, LOW);
61   Serial.println("no object found");
62   object="No";
63 }
64 String payload="{\"distance\": ";
65 payload+=dist;
66 payload+=", \"object\": \"";
67 payload+=object;
68 payload+="\"}";
69 Serial.print("Sending payload: ");
70 Serial.println(payload);
71 if(client.publish(publishTopic, (char*) payload.c_str())){
72   Serial.println("Publish ok");
73 }
74 else
75 {
76   Serial.println("Publish failed");
77 }
78 }
79 void mqttConnect()
80 {
81   if(!client.connected()){
```

Wokwi IDE interface showing the code for `esp32-blink.ino`. The code is as follows:

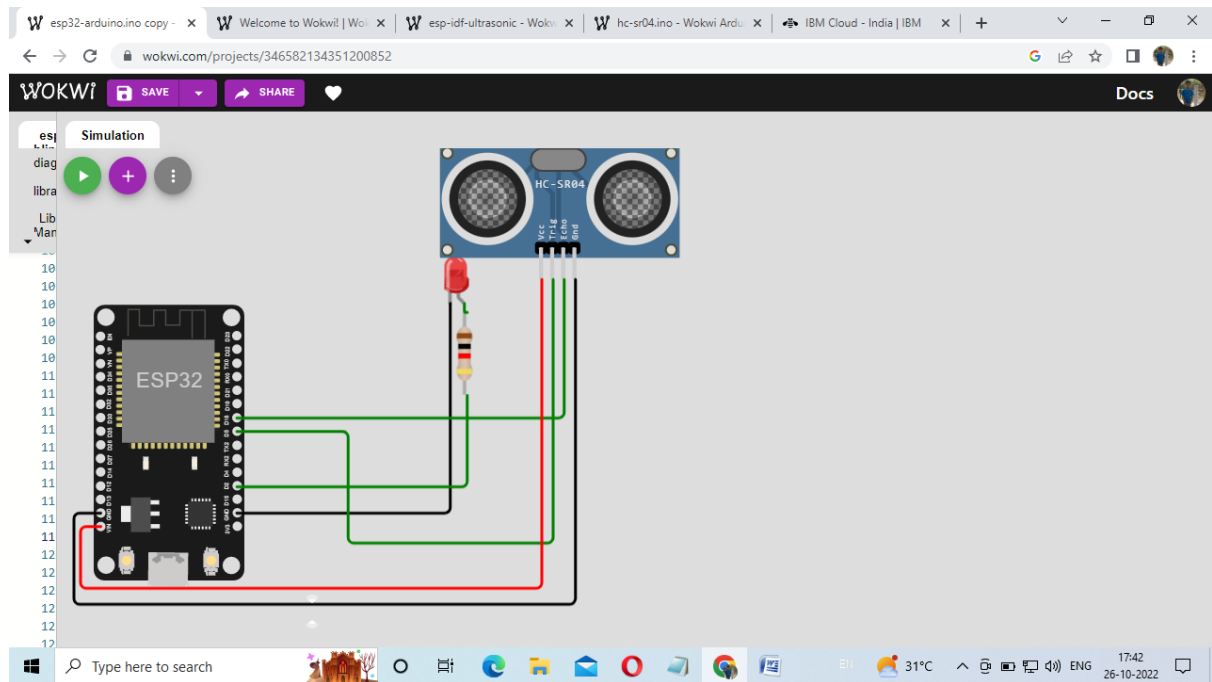
```
80 {
81   if(!client.connected()){
82     Serial.print("Reconnecting client to");
83     Serial.println(server);
84     while(!client.connect(clientId,authMethod,token)){
85       Serial.print(".");
86       delay(500);
87     }
88     initManagedDevice();
89     Serial.println();
90   }
91 }
92 void wificonnect()
93 {
94   Serial.println();
95   Serial.print("Connecting to");
96   Wifi.begin("Wokwi-GUEST", "",6);
97   while(Wifi.status()!=WL_CONNECTED){
98     delay(500);
99     Serial.print(".");
100   }
101   Serial.println("");
102   Serial.println("Wifi connected");
103   Serial.println("IP address");
104   Serial.println(Wifi.localIP());
105 }
106 void initManagerDevice(){
107   if(client.subscribe(subscribetopic)){
```

The interface includes a top bar with "WOKWI", "SAVE", "SHARE", and "Docs" buttons. The left sidebar shows "esp32-blink.ino", "diagram.json", "libraries.txt", and "Library Manager". The right sidebar shows a "Sim" button and a circuit diagram. The bottom status bar shows the Windows taskbar with a search bar, system icons, and the date/time "17:35 26-10-2022".

Wokwi IDE interface showing the code for `esp32-blink.ino`. The code is as follows:

```
104   Serial.println(Wifi.localIP());
105 }
106 void initManagerDevice(){
107   if(client.subscribe(subscribetopic)){
108     Serial.println(subscribetopic);
109     Serial.println("subscribe to cmd OK");
110   }
111   else
112   {
113     Serial.println("subscribe to cmd FAILED");
114   }
115 }
116 void callback(char* subscribetopic,byte* payload,unsigned int payloadLength)
117 {
118   Serial.print("callback invoked for topic:");
119   Serial.println(subscribetopic);
120   for(int i=0;i<payloadLength;i++)
121   {
122     data3+=(char)payload[i];
123   }
124   data3="";
125 }
```

The interface includes a top bar with "WOKWI", "SAVE", "SHARE", and "Docs" buttons. The left sidebar shows "esp32-blink.ino", "diagram.json", "libraries.txt", and "Library Manager". The right sidebar shows a "Sim" button and a circuit diagram. The bottom status bar shows the Windows taskbar with a search bar, system icons, and the date/time "17:35 26-10-2022".



IBM WATSON APP ISSUES ONLY WOKWI SIMULATOR ATTACHED.