ASSIGNMENT-- 4

Industry-specific intelligent fire management system

Team id : PNT2022TMID29717

Question :

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

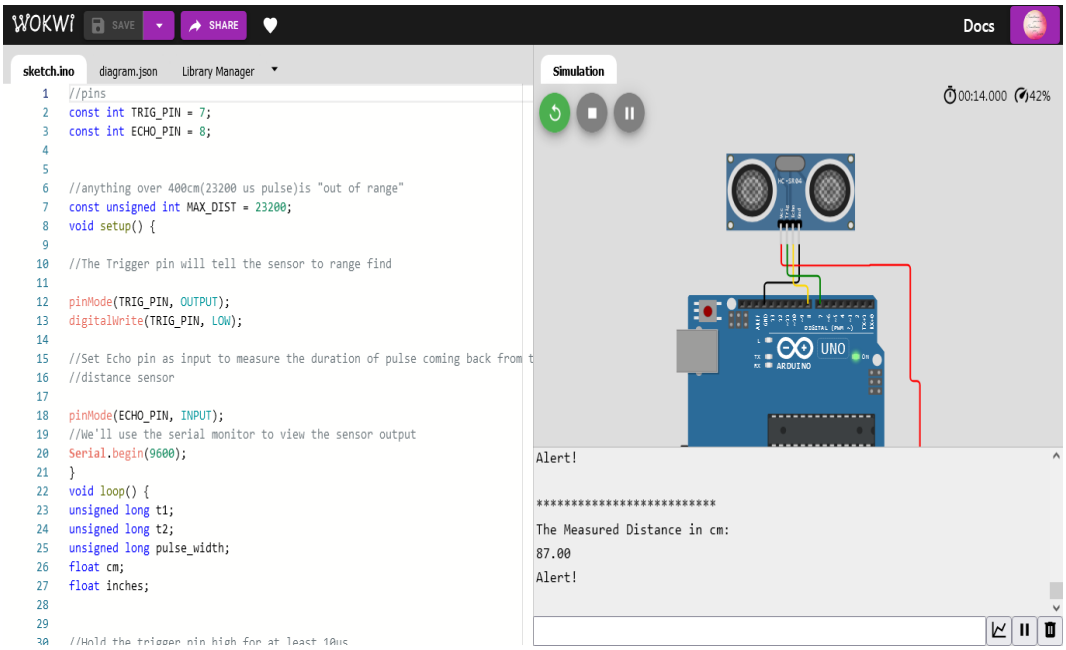
❖ Code :

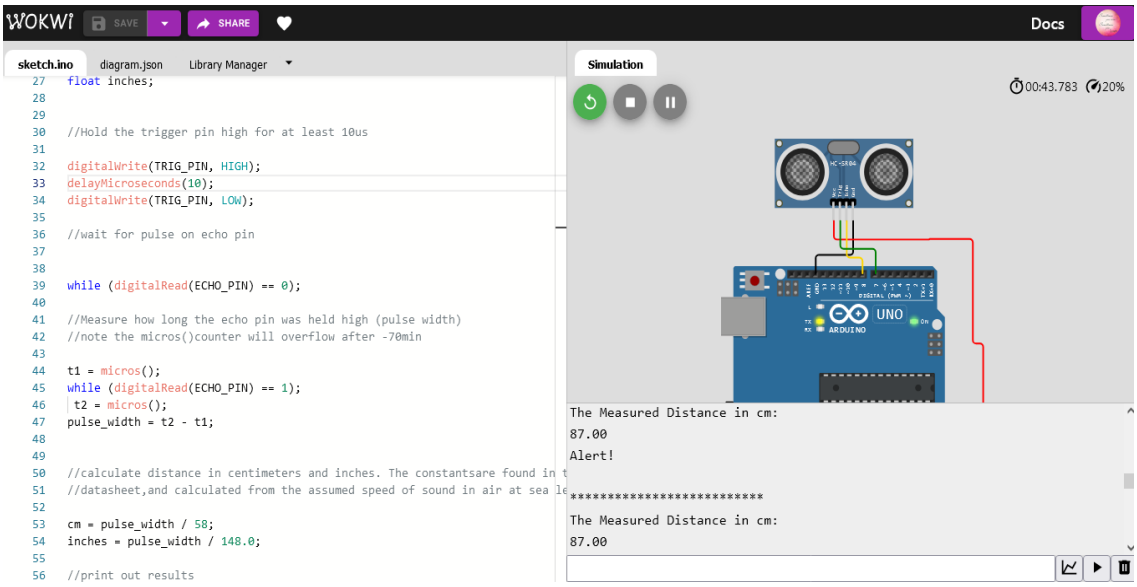
```
#include <WiFi.h>
#include <PubSubClient.h>

#define ORG "0bm892"
#define DEVICE_TYPE "ESP32_Controller"
#define DEVICE_ID "Sensor"
#define TOKEN "1234567890"
#define trigpin 5
#define echopin 18
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
long duration;
float dist;
void setup()
{
 Serial.begin(9900);

 pinMode(trigpin, OUTPUT);
 pinMode(echopin, INPUT);
 wifiConnect();
 mqttConnect();

}
void loop() {

 publishData();
 delay(500);
 if (!client.loop())
 {
 mqttConnect();
 }
```

```
}
void wifiConnect()
{
 Serial.print("Connecting to ");
 Serial.print("Wifi");
 WiFi.begin("Wokwi-GUEST", "", 6);
 while (WiFi.status() != WL_CONNECTED)
 {
 delay(500);
 Serial.print(".");
 }
 Serial.print("WiFi connected, IP address: ");
 Serial.println(WiFi.localIP());

}
void mqttConnect()
{
 if (!client.connected())
 {
 Serial.print("Reconnecting MQTT client to ");
 Serial.println(server);
 while (!client.connect(clientId, authMethod, token))
 {
 Serial.print(".");
 delay(500);
 }
  Serial.println();
 }
}
void publishData()
{
 digitalWrite(trigpin,LOW);
 digitalWrite(trigpin,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigpin,LOW);
 duration=pulseIn(echopin,HIGH);
 dist=(duration*0.034) /2;
 if(dist<100)
 {
 String payload = "{\"Distance\":";
 payload += dist;
 payload += ",";
 payload += "\"Status\":";
 payload += "\"Alert\"}";
 Serial.print("\n");
 Serial.print("Sending payload: ");
 Serial.println(payload);
 if (client.publish(publishTopic, (char*) payload.c_str()))
```

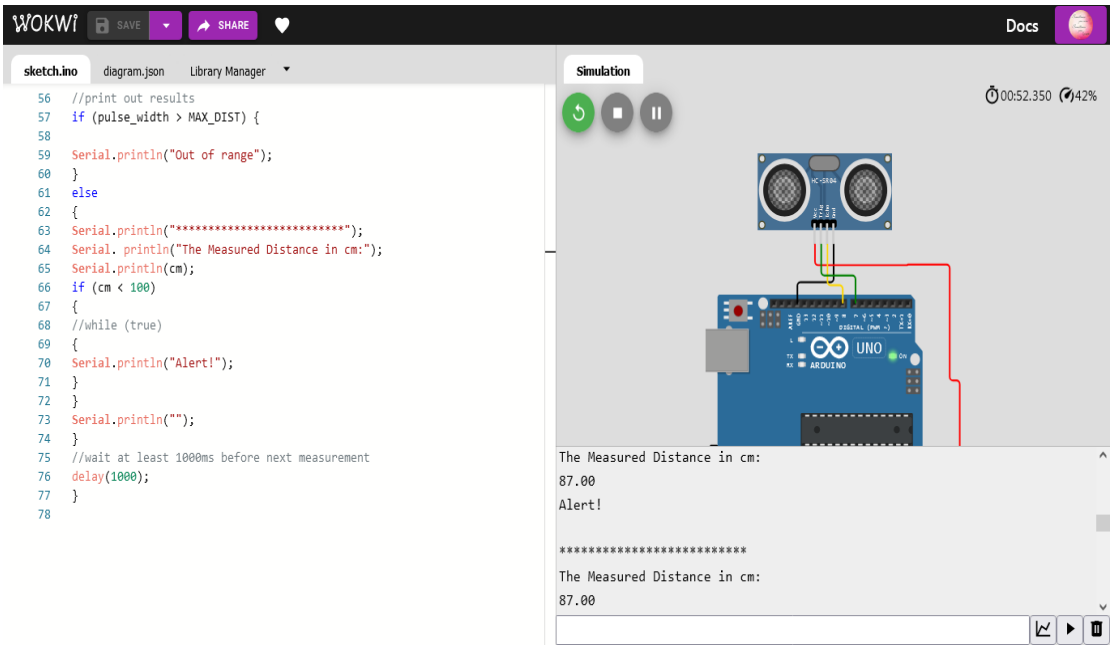
```
{
 Serial.println("Publish OK");
 }
 }
 if(dist>100)
 {
 String payload = "{\"Distance\":";
 payload += dist;
 payload += ",";
 payload += "\"Status\":";
 payload += "\"Normal\"}";
 Serial.print("\n");
 Serial.print("Sending payload: ");
 Serial.println(payload);
 if(client.publish(publishTopic, (char*) payload.c_str()))
 {
 Serial.println("Publish OK");
 }
 else
 {
    Serial.println("Publish FAILED");
 }
 }
 }
```

❖ Execution :



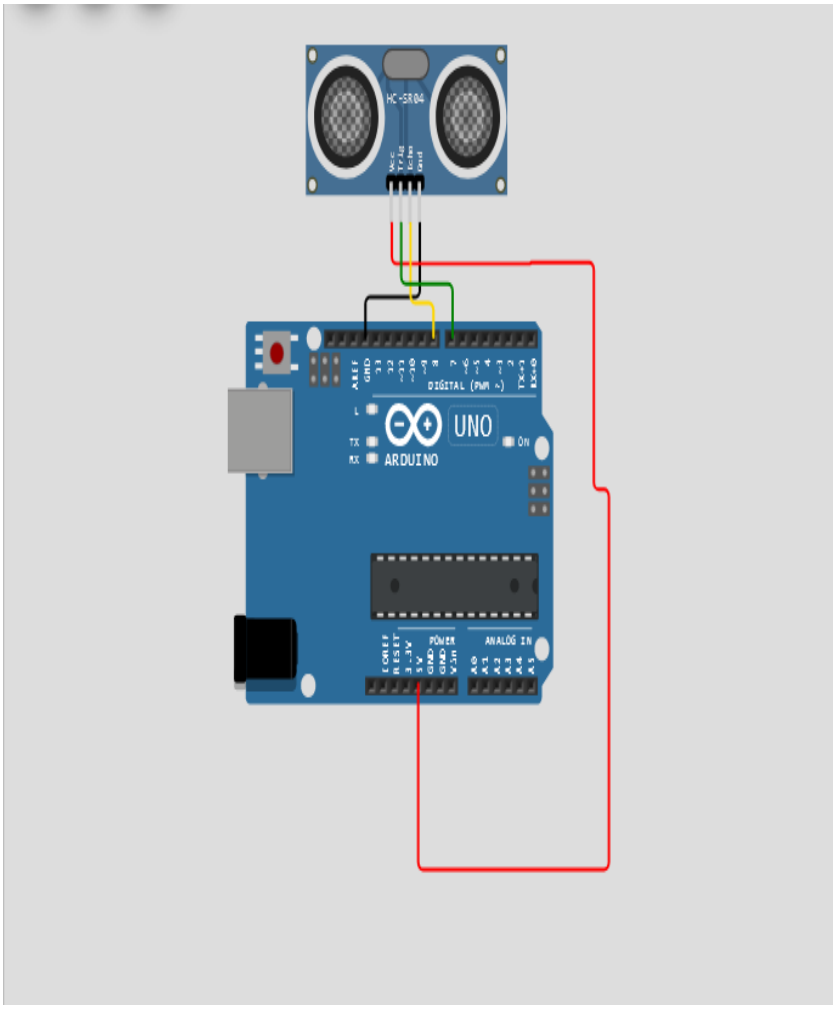



```
float inches;


//Hold the trigger pin high for at least 10us

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

//wait for pulse on echo pin


while (digitalRead(ECHO_PIN) == 0);

//Measure how long the echo pin was held high (pulse width)
//note the micros()counter will overflow after ~70min

t1 = micros();
while (digitalRead(ECHO_PIN) == 1);
 t2 = micros();
pulse_width = t2 - t1;


//calculate distance in centimeters and inches. The constantsare found in t
//datasheet,and calculated from the assumed speed of sound in air at sea le

cm = pulse_width / 58;
inches = pulse_width / 148.0;

//print out results
```

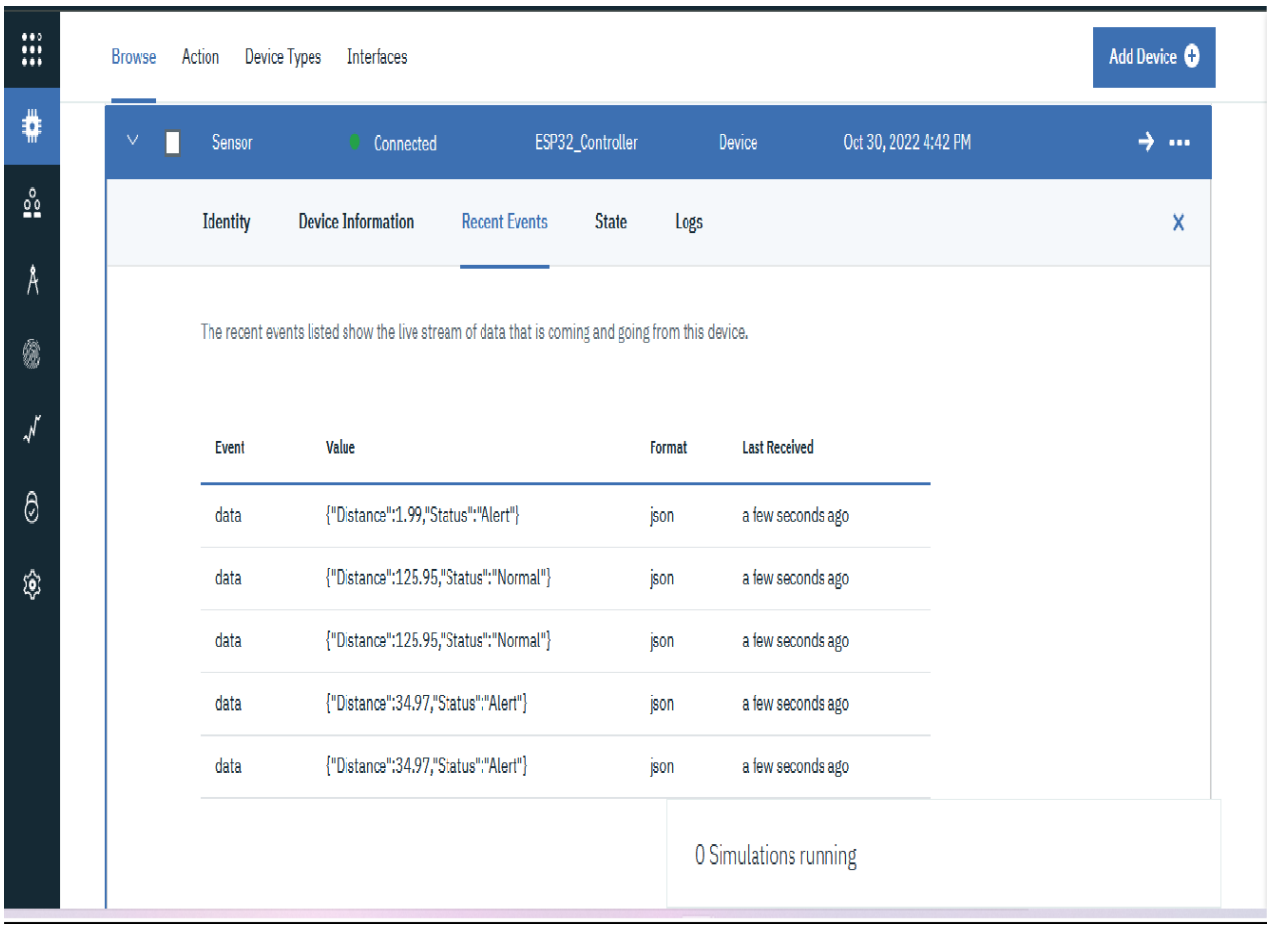



```
//print out results
if (pulse_width > MAX_DIST) {

Serial.println("Out of range");
}
else
{
Serial.println("**********");
Serial. println("The Measured Distance in cm:");
Serial.println(cm);
if (cm < 100)
{
//while (true)
{
Serial.println("Alert!");
}
}
Serial.println("");
}
//wait at least 1000ms before next measurement
delay(1000);
}

```

❖ **circuit diagram :**



IBM cloud output :



- **Wokwi URL : https://wokwi.com/projects/346948920308400723**