```
#include "SSD1306Wire.h"

#include <WiFi.h>

#include <WiFiMulti.h>

#include <FirebaseESP32.h>


WiFiMulti wifiMulti;


////////////////////////////OLED////////////////////////

#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define SDA 21

#define SCL 22




//LED PIN

#define GREEN        25

#define RED        26

#define YELLOW       27


#define BUZZER_PIN        12




///////////////ULTRASONIC_SENSOR/////////////////////////

const int trigPin = 5;

const int echoPin = 18;




// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

SSD1306Wire display(0x3c, SDA, SCL);
```

```
//////////////////////////////FIREBASE/////////////////////////////
#define FIREBASE_HOST "https://esp32andfirebase-36454-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "qAFt9dUyDhj5zd62N2VbeOM10LSYrGZdncDqTHmm"
//#define WIFI_SSID "Tenda_B5D145_5G"
//#define WIFI_PASSWORD "bodycalm485"
//Define FirebaseESP32 data object
FirebaseData firebasedata;
FirebaseJson json;



//define sound speed in cm/uS
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701

long duration;
int distanceCm;
int distanceInch;
int range = 20;//range of max water distance
//prototype
void ultrasonic();
void oled();
void hantar ();



void setup() {
  Serial.begin(115200);

  delay(10);

      wifiMulti.addAP("saldilah@unifi", "820304105332");
      wifiMulti.addAP("AK", "farhannoob");
```

```
   wifiMulti.addAP("Akmal", "msu123456789");
   wifiMulti.addAP("$.DURKA*", "vkqt7155");




       Serial.println("Connecting Wifi...");
   //WiFi.begin();
   if(wifiMulti.run() == WL_CONNECTED) {
      Serial.println("");
      Serial.println("WiFi connected to");
      Serial.println(WiFi.SSID());
      Serial.println("IP address: ");
      Serial.println(WiFi.localIP());
   }

     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

//Set database read timeout to 1 minute (max 15 minutes)
Firebase.setReadTimeout(firebasedata, 1000*60);
//tiny, small, medium, large and unlimited.
//Size and its write timeout e.g. tiny (1s), small (10s), medium (30m) and large (60m).
Firebase.setwriteSizeLimit(firebasedata, "tiny");



 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  // Initialising the UI will init the display too.
   display.init();
```

```
    display.flipScreenVertically();
    display.setFont(ArialMT_Plain_10);
  //INITIALIZE LED AND ALARM//
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(YELLOW, OUTPUT);

  //WIFI//


}

void loop() {

  if(wifiMulti.run() != WL_CONNECTED) {
        Serial.println("WiFi not connected!");
        delay(1000);
    }

  delay(1000);

  ultrasonic();
  delay(100);
 // clear the display
  display.clear();
  oled();
  display.display();
  delay(100);
  hantar();
```

```
}




void ultrasonic(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(12);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance
  distanceCm = duration * SOUND_SPEED/2;

  // Convert to inches
  distanceInch = distanceCm * CM_TO_INCH;

  // Prints the distance in the Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);
  Serial.print("Distance (inch): ");
  Serial.println(distanceInch);

  //Alarm and LED configuration
  if(distanceCm <= 2){
    Serial.println("DANGER!!!");
    digitalWrite(RED, HIGH);
    digitalWrite(GREEN, LOW);
    digitalWrite(YELLOW, LOW);
    digitalWrite(BUZZER_PIN, HIGH);
    delay(100);
  }
  else if (distanceCm < 6 && distanceCm > 3){
    Serial.println("CAUTION!!");
    digitalWrite(YELLOW,HIGH);
    digitalWrite(GREEN, LOW);
    digitalWrite(RED, LOW);
    delay(500);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(YELLOW, LOW);
    digitalWrite(GREEN, LOW);
    digitalWrite(RED, LOW);
    delay(100);
  }else {
    Serial.println("GOOD : ]");
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(GREEN,HIGH);
    digitalWrite(RED, LOW);
    digitalWrite(YELLOW, LOW);
    delay(100);
  }

}

void hantar (){
```

```
int Sdata = distanceCm;


  delay (100);
  json.set("/distance", Sdata);
  json.set("/Location", "MSU");
  Firebase.updateNode(firebasedata, "/water Level", json);


  if(distanceCm <= 3){
    json.set("/warning","DANGER!!! ");
    Firebase.updateNode(firebasedata, "/water Level", json);
    delay(100);
  } else if (distanceCm < 9 && distanceCm > 3){
    json.set("/warning","CAUTION!!");
    Firebase.updateNode(firebasedata, "/water Level", json);
    delay(100);
  }else{
    json.set("/warning","GOOD :]");
    Firebase.updateNode(firebasedata, "/water Level", json);
    delay(100);
  }


}
void oled(){
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.setFont(ArialMT_Plain_16);
  display.drawString(0, 0, "Distance in CM");



  String myString = String(distanceCm);


  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.setFont(ArialMT_Plain_16);
  display.drawString(20, 20, myString);



  if(distanceCm <= 3){
    display.setTextAlignment(TEXT_ALIGN_CENTER);
    display.setFont(ArialMT_Plain_10);
    display.drawString(40, 45, "DANGER!!!");

  } else if(distanceCm < 9 && distanceCm > 3){
    display.setTextAlignment(TEXT_ALIGN_CENTER);
    display.setFont(ArialMT_Plain_10);
    display.drawString(60, 45,"CAUTION!!" );
  }else{
    display.setTextAlignment(TEXT_ALIGN_CENTER);
    display.setFont(ArialMT_Plain_10);
    display.drawString(60, 45,"GOOD : ]" );
  }

}
```