
UUM
Universiti Utara Malaysia

**SCHOOL OF COMPUTING**
**UUM COLLEGE OF ARTS AND SCIENCES**

**STTHK3113 SENSOR-BASED SYSTEMS (A)**
**SEMESTER 6 (A242)**

# FINAL PROJECT ASSIGNMENT: SMART SENSOR-BASED SYSTEM

AQUASENTRY: SMART SINK WATER WASTE & LEAK DEFENDER

**PREPARED FOR :**
**AHMAD HANIS BIN MOHD SHABLI**

**PREPARED BY**


| NAME | MATRIC NO. |
|---|---|
| MOHAMMED UMAIR BIN MOHAMMED SUHAIMEE | 295498 |


**SUBMISSION DATE: 19TH JUNE 2025**

## 1.0 PROJECT TITLE

AquaSentry: Smart Sink Water Waste & Leak Defender

## 1.2 PROJECT DESCRIPTION

*AquaSentry* is a smart IoT-based system designed to prevent water wastage and detect leak incidents in real time at sink areas. This system will utilize an **ultrasonic sensor** mounted above the sink to detect the presence or absence of hands. If water is running and no hands are detected for a prolonged period, the system will classify the event as water waste alert. Simultaneously, a **water leak sensor** will be installed at the bottom of the sink or within the cabinet to detect any sign of dripping or leakage from broken pipes or fittings. If either condition is met, a relay module will be triggered to alert the user immediately. Sensor readings will be collected every second for temperature, humidity, water, and ultrasonic detection.

In addition to the hardware system, an **interactive real-time dashboard website** will be built to visually present sensor data using live graphs, sensor states, and event logs. All readings are logged into a database and categorized by sensor type, including temperature, humidity, ultrasonic, water level, and relay state. Users can view the status via mobile, desktop, or tablet, and filter logs or zoom into data using intuitive UI features. The system ensures efficient monitoring, precise real-time feedback, and proactive alerts — making AquaSentry a practical solution for smarter water management at any household or facility.

## 1.3 OBJECTIVE

1. To implement and integrate multiple sensors (ultrasonic, water, DHT11) on the ESP32 to detect water leaks and water waste conditions.
2. To build a real-time interactive website that visualizes all sensor readings, relay states, and system logs for public monitoring.
3. To reduce unnecessary water usage and raise awareness through automated detection and user-friendly visualization of water activities at the sink.

## 2.0 CIRCUIT DIAGRAM

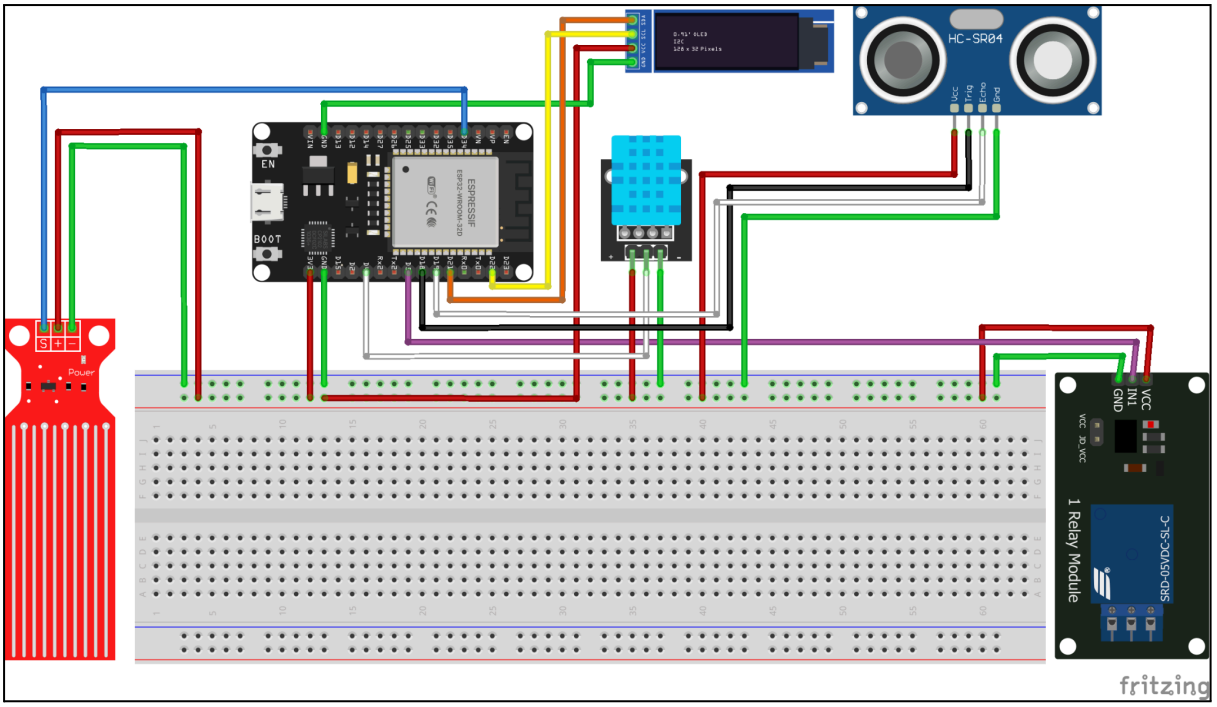



***Figure 1.*** *AquaSentry Circuit Diagram – Fritzing*

***Table 1.*** *ESP32 Pin Connection*

| **ESP32** | **Connection** |
|---|---|
| GPIO4 | DHT11 DATA |
| GPIO5 | Relay IN |
| GPIO18 | Ultrasonic TRIG |
| GPIO19 | Ultrasonic ECHO |
| GPIO21 (SDA) | OLED (SDA) |
| GPIO22 (SCL) | OLED (SCK) |
| GPIO34 | Water Sensor OUT (analog) |

## 3.0 ARDUINO CODE

```
#include <Wire.h>
#include <WiFi.h>
#include <Preferences.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

// ---------- PIN DEFINITIONS ----------
#define TRIG_PIN    18
#define ECHO_PIN    19
#define LEAK_PIN    34
#define RELAY_PIN   5
#define DHT_PIN     4
#define DHT_TYPE    DHT11

// ---------- OLED SETUP ----------
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Only redraw when something really changes
bool displayDirty = false;

// ---------- SCROLLING TEXT STRUCTURE ----------
struct ScrollLine {
  String text;
  int y;
  int offset;
  int scrollDir;
  int scrollRange;
  int scrollSpeed;
  unsigned long lastReverse;
  bool waiting;
};
ScrollLine lines[4];
unsigned long lastScrollUpdate = 0;
const int SCROLL_INTERVAL = 30;

// ---------- WIFI VIA PREFERENCES ----------
Preferences preferences;
#define MAX_WIFI 3
String ssidList[MAX_WIFI], passList[MAX_WIFI], currentSSID;

// ---------- DHT SENSOR ----------
DHT dht(DHT_PIN, DHT_TYPE);

// ---------- TIMING & STATE ----------
unsigned long lastPost = 0;
const unsigned long POST_INTERVAL = 1000;
int lastRelayState = LOW;
```

```
float lastDistValue = -1;
unsigned long stableSince = 0;

// ---------- FUNCTION PROTOTYPES ----------
void displayStatus(const String &msg);
void loadWiFiCredentials();
void connectToWiFi();
float readDistance();
int readWaterLevel();
bool getStableDHTReading(float &temp, float &hum);
void postSensorData(const String &eventType, float dist, int water, bool
leak, float temp, float hum);
void postRelayState(int state);

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

  // ← Initialize I2C on GPIO21=D21 (SDA) and GPIO22=D22 (SCL)
  Wire.begin(21, 22);

  // OLED init
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 allocation failed");
    for(;;);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  for (int i = 0; i < 4; i++) {
    lines[i] = {"", i * 8, 0, -1, 0, 1, 0, false};
  }

  // pins
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(LEAK_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);

  // sensors
  dht.begin();

  // Wi-Fi
  loadWiFiCredentials();
  connectToWiFi();
}

void loop() {
  // reconnect if dropped
  if (WiFi.status() != WL_CONNECTED) {
    connectToWiFi();
  }
```

```
unsigned long now = millis();

// read sensors
float dist = readDistance();
int waterLevel = readWaterLevel();           // raw analog value
bool leak = (waterLevel > 1500);             // threshold
float temp, hum;
getStableDHTReading(temp, hum);

// determine presence
bool presence;
if (lastDistValue < 0) {
  lastDistValue = dist;
  stableSince = now;
  presence = false;
} else if (dist != lastDistValue) {
  presence = true;
  lastDistValue = dist;
  stableSince = now;
} else {
  presence = (now - stableSince) < POST_INTERVAL * 10;
}

// relay logic: ON if leaking OR no presence; OFF otherwise
int newRelay = (leak || !presence) ? HIGH : LOW;
if (newRelay != lastRelayState) {
  digitalWrite(RELAY_PIN, newRelay);
  postRelayState(newRelay);
  lastRelayState = newRelay;
}

// post sensor data every second
if (now - lastPost >= POST_INTERVAL) {
  String eventType = leak ? "leak_detected"
                     : (!presence ? "waste_alarm" : "presence");
  postSensorData(eventType, dist, waterLevel, leak, temp, hum);
  lastPost = now;
}

// prepare display lines
String L1 = "UltSonic : " + String(dist,1) + " cm";
String L2 = "Water    : " + String(waterLevel);
String L3 = "Presence : " + String(presence ? "Detected" : "No");
String L4 = "Leaking  : " + String(leak ? "Yes" : "No");

// check for text changes
String newLines[4] = {L1, L2, L3, L4};
bool changed = false;
for (int i = 0; i < 4; i++) {
  if (newLines[i] != lines[i].text) {
    changed = true;
    break;
  }
```

```
  }
  if (changed) {
    // recalc scroll ranges
    for (int i = 0; i < 4; i++) {
      lines[i].text = newLines[i];
      int16_t x1, y1; uint16_t w, h;
      display.getTextBounds(lines[i].text, 0, lines[i].y, &x1, &y1, &w, &h);
      lines[i].scrollRange = max(0, (int)w - SCREEN_WIDTH);
      lines[i].offset = 0;
      lines[i].scrollDir = -1;
      lines[i].waiting = false;
    }
    displayDirty = true;
  }

  // redraw/scroll only when dirty
  if (displayDirty && now - lastScrollUpdate >= SCROLL_INTERVAL) {
    lastScrollUpdate = now;
    display.clearDisplay();
    for (int i = 0; i < 4; i++) {
      auto &ln = lines[i];
      if (ln.scrollRange > 0 && !ln.waiting) {
        ln.offset += ln.scrollDir * ln.scrollSpeed;
        if (ln.offset <= -ln.scrollRange || ln.offset >= 0) {
          ln.scrollDir *= -1;
          ln.waiting = true;
          ln.lastReverse = now;
        }
      }
      if (ln.waiting && now - ln.lastReverse > 1000) {
        ln.waiting = false;
      }
      display.setCursor(ln.offset, ln.y);
      display.println(ln.text);
    }
    display.display();
    displayDirty = false;
  }

  delay(1000);
}

// ——— SUPPORT FUNCTIONS ———

void displayStatus(const String &msg) {
  display.clearDisplay();
  display.setCursor(0,0);
  display.println(msg);
  display.display();
  delay(1000);
}

void loadWiFiCredentials() {
```

```
  preferences.begin("wifiCreds", true);
  for (int i = 0; i < MAX_WIFI; i++) {
    ssidList[i] = preferences.getString(("ssid"+String(i)).c_str(), "");
    passList[i] = preferences.getString(("pass"+String(i)).c_str(), "");
  }
  preferences.end();
}

void connectToWiFi() {
  WiFi.disconnect(true);
  WiFi.mode(WIFI_STA);
  delay(100);
  displayStatus("WiFi: Scanning...");
  int n = WiFi.scanNetworks();
  if (n == 0) {
    displayStatus("WiFi: No Network");
    return;
  }
  for (int i = 0; i < MAX_WIFI; i++) {
    if (ssidList[i] == "") continue;
    for (int j = 0; j < n; j++) {
      if (WiFi.SSID(j) == ssidList[i]) {
        WiFi.begin(ssidList[i].c_str(), passList[i].c_str());
        displayStatus("WiFi: Connecting to " + ssidList[i]);
        int tries = 0;
        while (WiFi.status() != WL_CONNECTED && tries < 20) {
          delay(500);
          tries++;
        }
        if (WiFi.status() == WL_CONNECTED) {
          currentSSID = ssidList[i];
          displayStatus("WiFi: " + currentSSID);
          return;
        }
      }
    }
  }
  displayStatus("WiFi: No WiFi");
}

float readDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long d = pulseIn(ECHO_PIN, HIGH, 30000);
  return (d * 0.0343) / 2.0;
}

int readWaterLevel() {
  return analogRead(LEAK_PIN);
}
```

```
bool getStableDHTReading(float &temp, float &hum) {
  float t, h;
  for (int i = 0; i < 3; i++) {
    t = dht.readTemperature();
    h = dht.readHumidity();
    if (!isnan(t) && !isnan(h)) {
      temp = t; hum = h;
      return true;
    }
    delay(200);
  }
  temp = isnan(t) ? 0 : t;
  hum  = isnan(h) ? 0 : h;
  return false;
}

void postSensorData(const String &eventType, float dist, int water, bool
leak, float temp, float hum) {
  HTTPClient http;
  http.begin("https://umairsuhaimee.com/aqua-sentry/api/post_reading.php");
  http.addHeader("Content-Type","application/x-www-form-urlencoded");
  String body = "event_type=" + eventType
              + "&distance="   + String(dist,1)
              + "&leak="       + String(leak?1:0)
              + "&temperature="+ String(temp,1)
              + "&humidity="   + String(hum,1);
  http.POST(body);
  http.end();
}

void postRelayState(int state) {
  HTTPClient http;
  http.begin("https://umairsuhaimee.com/aqua-sentry/api/post_reading.php");
  http.addHeader("Content-Type","application/x-www-form-urlencoded");
  String body = "relay=" + String(state==HIGH?"ON":"OFF");
  http.POST(body);
  http.end();
}
```

## 4.0 MOBILE APP INTERFACE

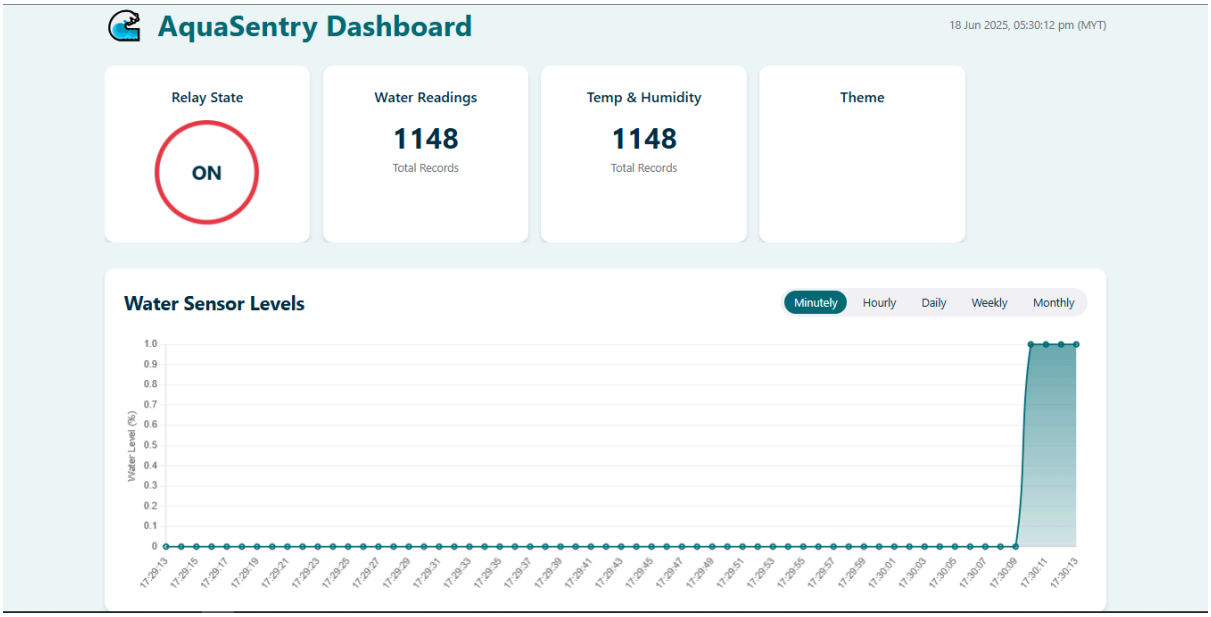



***Figure 2.*** *AquaSentry Dashboard Website – Status Cards and Water Sensor Graph*

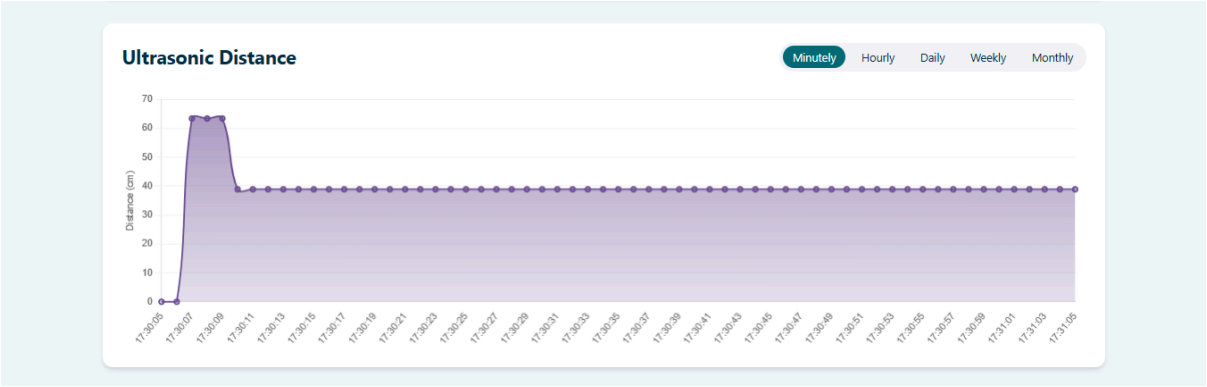



***Figure 3.*** *AquaSentry Dashboard Website – Ultrasonic Sensor Graph*

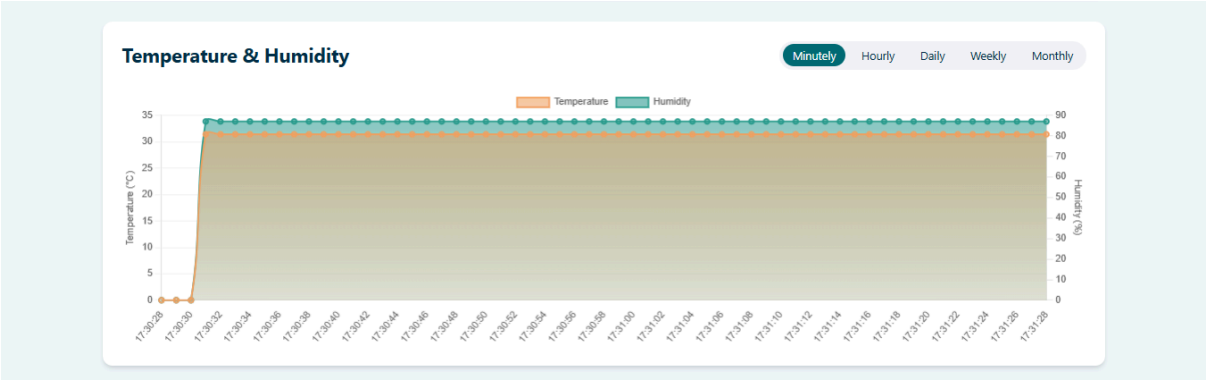
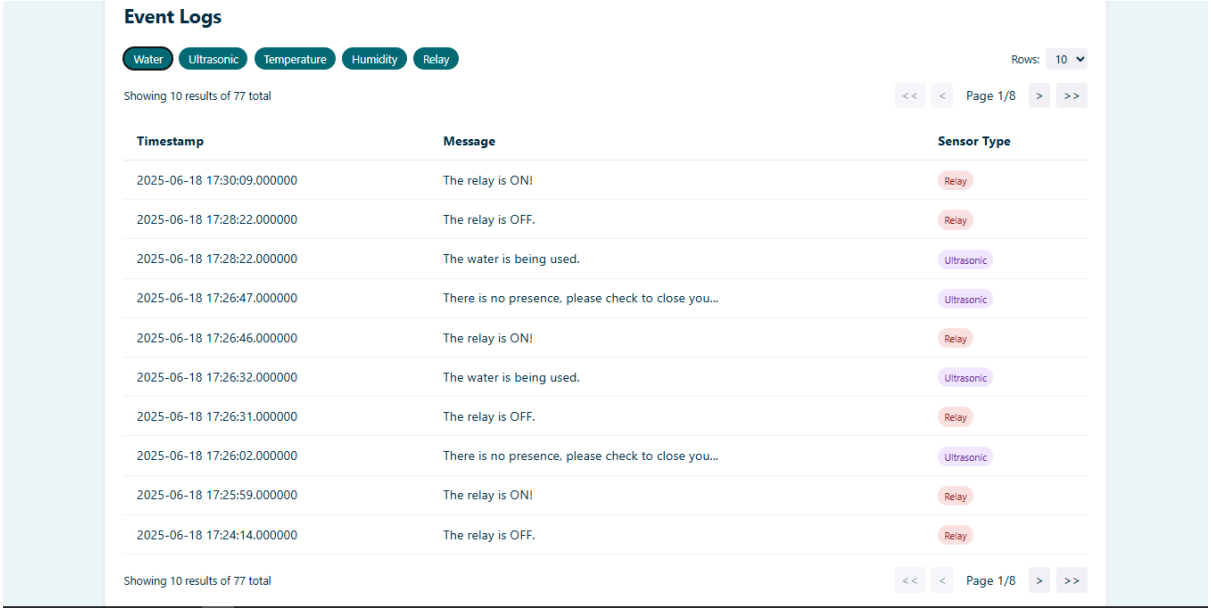



***Figure 4.*** *AquaSentry Dashboard Website – Temperature & Humidity Sensor Graph*



Event Logs

Water | Ultrasonic | Temperature | Humidity | Relay

Rows: 10

Showing 10 results of 77 total

<< < Page 1/8 > >>

| Timestamp | Message | Sensor Type |
|---|---|---|
| 2025-06-18 17:30:09.000000 | The relay is ON! | Relay |
| 2025-06-18 17:28:22.000000 | The relay is OFF. | Relay |
| 2025-06-18 17:28:22.000000 | The water is being used. | Ultrasonic |
| 2025-06-18 17:26:47.000000 | There is no presence, please check to close you... | Ultrasonic |
| 2025-06-18 17:26:46.000000 | The relay is ON! | Relay |
| 2025-06-18 17:26:32.000000 | The water is being used. | Ultrasonic |
| 2025-06-18 17:26:31.000000 | The relay is OFF. | Relay |
| 2025-06-18 17:26:02.000000 | There is no presence, please check to close you... | Ultrasonic |
| 2025-06-18 17:25:59.000000 | The relay is ON! | Relay |
| 2025-06-18 17:24:14.000000 | The relay is OFF. | Relay |

Showing 10 results of 77 total

<< < Page 1/8 > >>

***Figure 5.*** *AquaSentry Dashboard Website – Sensors Activity: Event Log*

## 5.0 YOUTUBE VIDEO

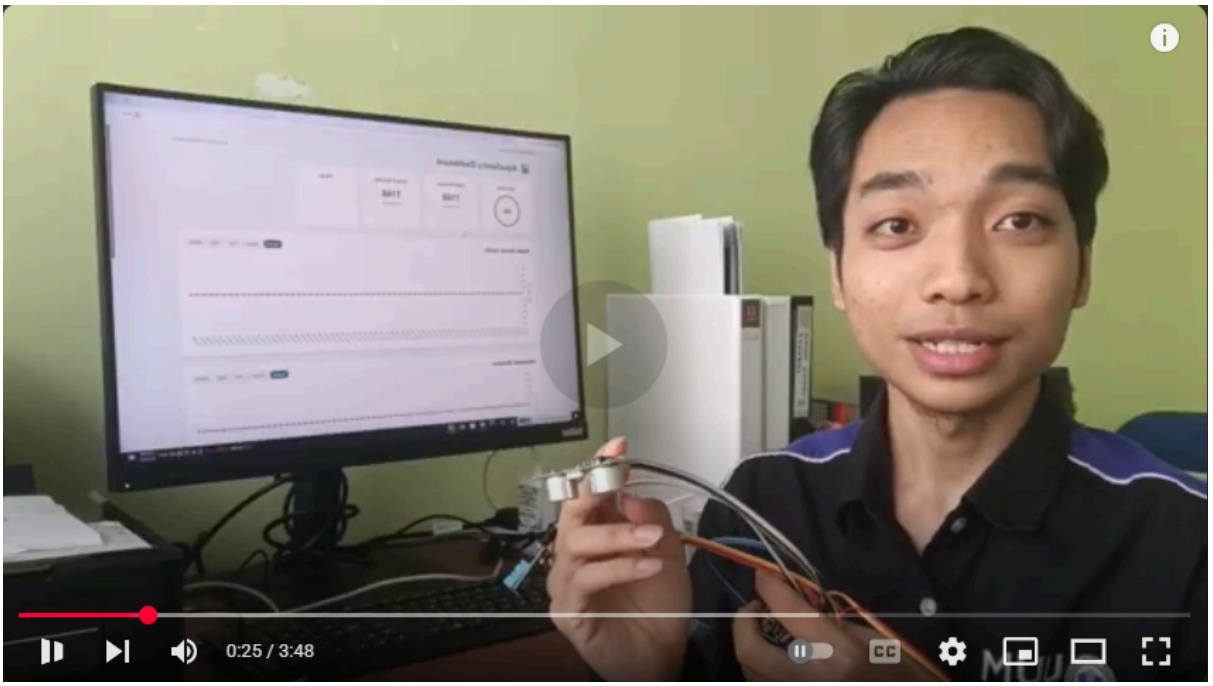



***Figure 6.*** *AquaSentry Demonstration Video – YouTube*

Link source: https://youtu.be/gdoY69vJKIg

## 6.0 GITHUB REPOSITORY

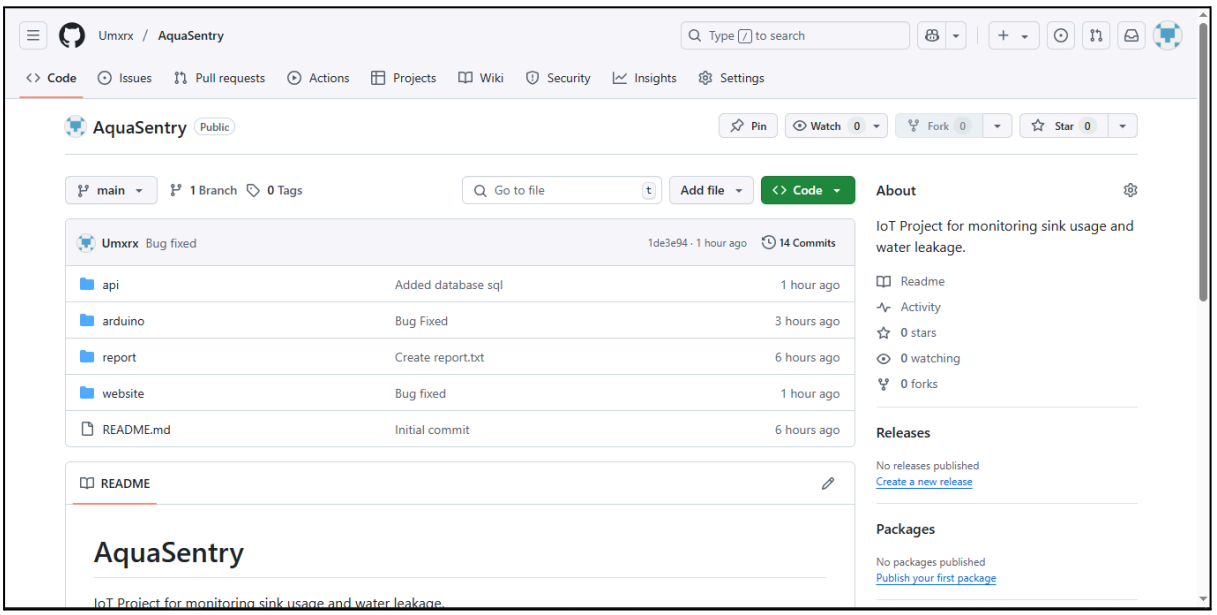



***Figure 7.*** *AquaSentry GitHub Repository*

Link source: https://github.com/Umxrx/AquaSentry

## 7.0 APPENDIX

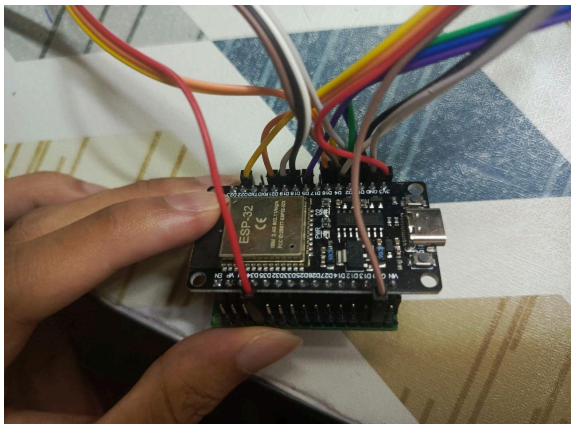



***Figure 8.*** *Arduino ESP32-WROOM-32D*

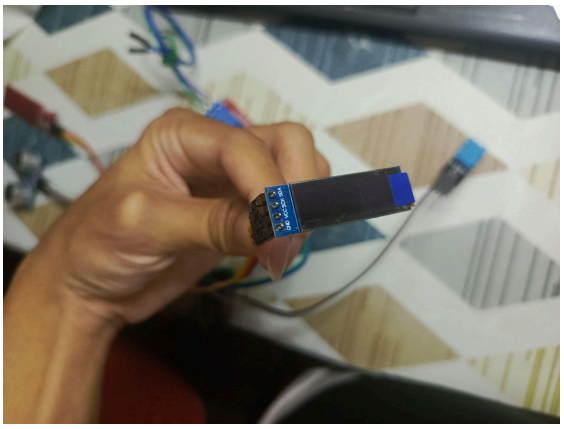


***Figure 9.*** *OLED Display 128px × 32px*

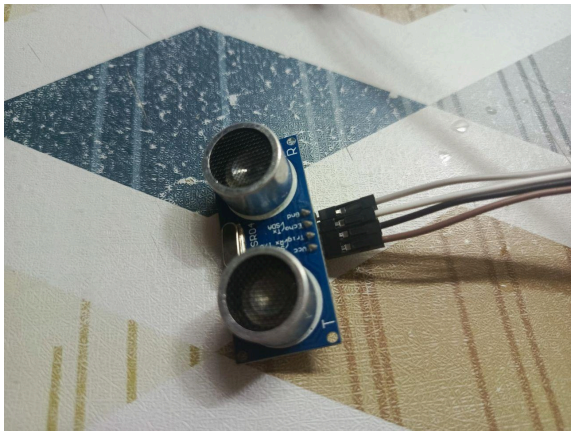


***Figure 10.*** *HC-SR04 Ultrasonic Sensor*


***Figure 11.*** *Water Level Sensor*

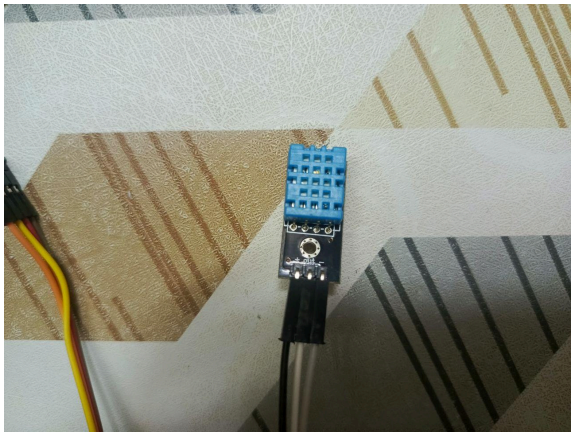


***Figure 12.*** *DHT11 Temperature & Humidity Sensor*

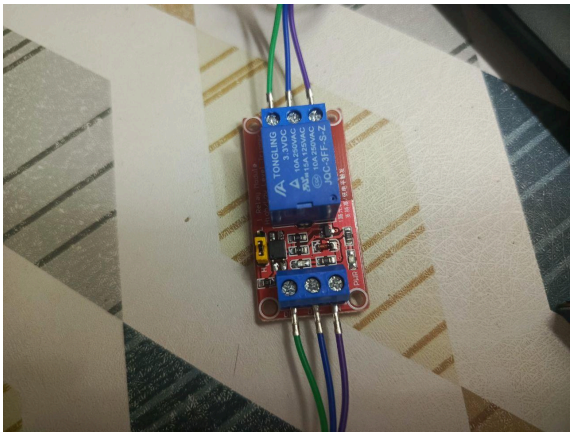



***Figure 13.*** *Optoisolated 1-Channel Relay*