Lab -02

Task-1 (Source code):

```
#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <DHT.h>


// OLED display width and height

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64


// OLED reset pin (usually-1)

#define OLED_RESET-1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


// DHT sensor settings

#define DHTPIN D4      // DHT11 data pin

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);


void setup() {

  Serial.begin(9600);


  // Initialize OLED

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found");
    while (true);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("DHT11 Test");
  display.display();
  delay(1000);

  // Initialize DHT sensor
  dht.begin();
}

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" C");

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  // Display on OLED
  display.clearDisplay();

  display.setCursor(0, 10);
  display.print("Temp: ");
  display.print(temperature);
  display.print(" C");

  display.setCursor(0, 30);
  display.print("Humidity: ");
  display.print(humidity);
  display.print(" %");

  display.display();
  delay(2000);
}
```

Task-3 (Source code):

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET-1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define RAIN_ANALOG A0   // Analog pin for rain sensor
int threshold = 600;     // Adjust after testing

void setup() {
  Serial.begin(9600);

  // Initialize OLED
  Wire.begin(D2, D1); // SDA, SCL for NodeMCU
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found");
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("Rain Sensor Ready");
  display.display();
  delay(1000);
}

void loop() {
  int val = analogRead(RAIN_ANALOG); // Read analog value
  String status;

  if (val < threshold) {
    status = "RAIN DETECTED ";
  } else {
    status = "NO RAIN ";
  }

  // Serial output
  Serial.print("Analog Value: ");
  Serial.print(val);
  Serial.print(" | Status: ");
  Serial.println(status);

  // OLED output
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Rain Sensor Monitor");
  display.setCursor(0, 25);
  display.print("Analog Value: ");
  display.println(val);

  display.setCursor(0, 50);
  display.println(status);

  display.display();

  delay(500);
}
```

Task-4 (Source code):

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
/*-------- OLED Config-------- */
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET-1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
/*-------- Pin Definitions-------- */
#define SOIL_PIN A0
#define BUZZER_PIN D5
/*-------- Sensor Calibration-------- */
```

```
// Put sensor in water → note value → assign to WET
// Put sensor in dry soil → note value → assign to DRY
const int wetValue = 300;   // Example: sensor value in wet soil
const int dryValue = 900;   // Example: sensor value in dry soil
void setup() {
  Serial.begin(9600);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);
  // NodeMCU I2C pins
  Wire.begin(D2, D1); // SDA, SCL
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not detected");
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  Serial.println("Soil Moisture Monitor Initialized!");
}

void loop() {
  int soilValue = analogRead(SOIL_PIN);

  // Clamp the value in case of floating pin
```

```
if (soilValue > 1023) soilValue = 1023;
if (soilValue < 0) soilValue = 0;

String status;

// Compare against calibrated thresholds
if (soilValue >= dryValue) {       // High value = Dry
  status = "DRY";
  digitalWrite(BUZZER_PIN, HIGH);
}
else if (soilValue <= wetValue) {  // Low value = Wet
  status = "WET";
  digitalWrite(BUZZER_PIN, HIGH);
}
else {
  status = "NORMAL";
  digitalWrite(BUZZER_PIN, LOW);
}

// Serial Monitor Output
Serial.print("Soil Value: ");
Serial.print(soilValue);
Serial.print(" | Status: ");
Serial.println(status);

// OLED Display Output
```

```
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Soil Moisture Monitor");

  display.setCursor(0, 20);
  display.print("ADC Value: ");
  display.println(soilValue);
  display.setCursor(0, 40);
  display.print("Status: ");
  display.println(status);
  display.display();
  delay(1000);
}
```

Task-5 (Source code):

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET-1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define WATER_PIN A0   // Analog pin for water level sensor
```

```
void setup() {
  Serial.begin(9600);

  // Initialize OLED
  Wire.begin(D2, D1); // SDA, SCL
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found");
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("Water Level Monitor");
  display.display();
  delay(1000);
}

void loop() {
  int waterValue = analogRead(WATER_PIN); // 0-1023

  // Convert to percentage
  int waterPercent = map(waterValue, 0, 1023, 0, 100);

  // Determine category
```

```
String level;
if (waterPercent < 30) {
  level = "LOW";
} else if (waterPercent <= 70) {
  level = "MEDIUM";
} else {
  level = "HIGH";
}

// Serial output
Serial.print("Water ADC: ");
Serial.print(waterValue);
Serial.print(" | Percent: ");
Serial.print(waterPercent);
Serial.print("% | Level: ");
Serial.println(level);

// OLED output
display.clearDisplay();
display.setCursor(0, 0);
display.println("Water Level Sensor");

display.setCursor(0, 25);
display.print("Level: ");
display.println(level);
```

```
  display.setCursor(0, 45);
  display.print("Percent: ");
  display.print(waterPercent);
  display.println("%");

  display.display();

  delay(500);
}
```

Task 6***- (Source code):

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "DHT.h"

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET-1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define LM35_PIN A0       // LM35 analog pin
#define DHTPIN D4         // DHT11 data pin
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();

  Wire.begin(D2, D1); // SDA, SCL

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found");
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("Temperature Monitor");
  display.display();
  delay(1000);
}

void loop() {
  //---- Read LM35----
  int lm35Value = analogRead(LM35_PIN); // 0-1023
  float voltage = lm35Value * (3.3 / 1023.0); // Convert ADC to voltage (V)
  float tempLM35 = voltage * 100.0;        // LM35: 10 mV per °C → 0.01V/°C, so 1V = 100°C
```

```
//---- Read DHT11----
float tempDHT = dht.readTemperature();
float humidity = dht.readHumidity();

//---- Serial output----
Serial.print("LM35 Temp: ");
Serial.print(tempLM35);
Serial.print(" °C | DHT11 Temp: ");
Serial.print(tempDHT);
Serial.print(" °C | Humidity: ");
Serial.print(humidity);
Serial.println(" %");

//---- OLED output----
display.clearDisplay();
display.setCursor(0, 0);
display.println("Temperature Monitor");

display.setCursor(0, 20);
display.print("LM35: ");
display.print(tempLM35);
display.println(" C");

display.setCursor(0, 35);
display.print("DHT11: ");
```

```
  display.print(tempDHT);
  display.println(" C");

  display.setCursor(0, 50);
  display.print("Humidity: ");
  display.print(humidity);
  display.println("%");

  display.display();

  delay(200);
}
```

Task 7- (Source code):

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "DHT.h"
#include "RTClib.h"

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET-1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

```
  }

#define DHTPIN D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

RTC_DS3231 rtc; // RTC module

void setup() {
  Serial.begin(9600);
  dht.begin();
  Wire.begin(D2, D1); // SDA, SCL

  // Initialize OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found");
    while (1);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  // Initialize RTC
  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  // Only set the time **once**, then comment it out
  rtc.adjust(DateTime(2026, 1, 20, 14, 21, 0)); // 20-01-2026 2:17 PM
}

void loop() {
  //---- Read DHT11----
  float temp = dht.readTemperature();
  float humidity = dht.readHumidity();

  //---- Read RTC----
  DateTime now = rtc.now();

  //---- Format date DD-MM-YYYY----
  char dateStr[11];
  sprintf(dateStr, "%02d-%02d-%04d", now.day(), now.month(), now.year());

  //---- Format time in 12-hour format with AM/PM----
  int hour = now.hour();
  char period[] = "AM";
  if(hour == 0) hour = 12;
  else if(hour == 12) period[0] = 'P';
  else if(hour > 12) { hour-= 12; period[0] = 'P'; }

  char timeStr[9];
  sprintf(timeStr, "%02d:%02d:%02d %s", hour, now.minute(), now.second(), period);
```

```
//---- Serial Output----
Serial.print(dateStr);
Serial.print(" ");
Serial.print(timeStr);
Serial.print(" | Temp: ");
Serial.print(temp);
Serial.print("°C | Humidity: ");
Serial.print(humidity);
Serial.println("%");

//---- OLED Output----
display.clearDisplay();
display.setCursor(0, 0);
display.println("Environment Monitor");

display.setCursor(0, 15);
display.print("Date: ");
display.println(dateStr);

display.setCursor(0, 30);
display.print("Time: ");
display.println(timeStr);

display.setCursor(0, 45);
display.print("Temp: ");
```

```
  display.print(temp);
  display.println(" C");

  display.setCursor(0, 60);
  display.print("Humidity: ");
  display.print(humidity);
  display.println(" %");

  display.display();
  delay(1000);
}
```