## - **Parts used :**

- (1x) Full+ Breadboard
- Jumper Wires
- (1x) Arduino Uno R3
- (1x) LCD (16x2)
- (1x) Blue LED
- (1x) Resistor 220ohm
- (2x) Potentiometer 10k
- (1x) Relay
- (1x) Water Level Detection Sensor
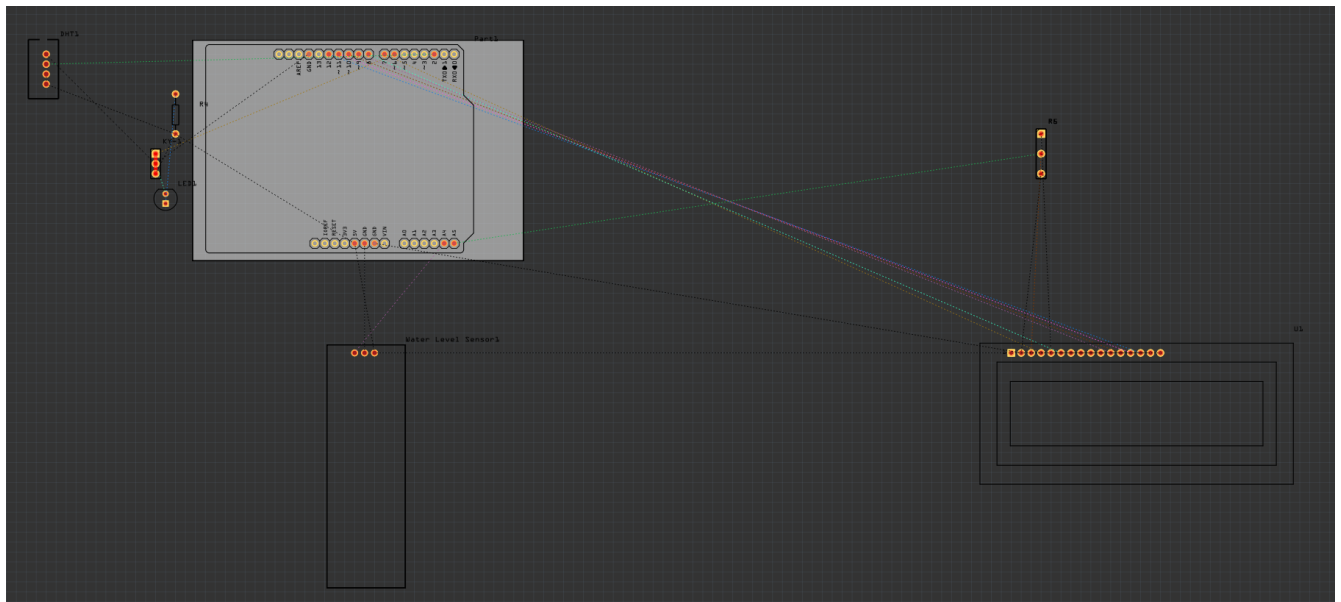- (1x) DHT11 Temperature and Humidity Sensor

## - **Software used :**

- Arduino – IDE

# Project Description :

The system is designed to monitors temperature and humidity levels in your room or environment. If the relative humidity falls below a required limit, the system activates a humidifier for 30 seconds and then rechecks the humidity level.

If it's still inadequate, the humidifier is turned on again.

# Algorithm and Program source code:

A) Algorithm :

**Initialize Components:**

- **Set up the LCD display and pins for sensors and actuators.**

- **Define variables for sensor readings ('temperature', 'humidity'), sensor pins ('pinDHT11', 'water', 'H'), output pin for the relay ('relay'), and variables to store sensor readings ('val', 'H_in', 'out').**

**Setup Function:**

- **Begin serial communication for debugging.**

- **Initialize the LCD display.**

- **Set pin modes for the relay and water sensor.**

**Main Loop (loop Function):**

- **Read temperature and humidity from the DHT11 sensor.**

- **Calculate humidity percentage from analog sensor readings.**

- **Display temperature and humidity on the LCD.**

- **Check if the current humidity is below a desired threshold (desiredHumidityLevel).**

- **If Below Threshold:**

  - **Turn on the humidifier (relay) for 30 seconds.**

  - **Read temperature and humidity again after 30 seconds.**

  - **Calculate humidity percentage.**

  - **If the humidity is now within the desired range, turn off the humidifier.**

**If Above or Within Threshold:**

- **Turn off the humidifier.**

**End of Loop:**

- **Repeat the loop.**

## B) Program Source Code :

```
#include <LiquidCrystal.h>
#include <SimpleDHT.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
int pinDHT11 = 2;
SimpleDHT11 dht11;
int water = A4;
int val = 0;
int H = A5;
int out = 0;
unsigned long H_in = 0;
int relay = 6;

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(relay, OUTPUT);
  pinMode(water, INPUT);
}

void loop() {
  byte temperature = 0;
  byte humidity = 0;
  byte data[40] = {0};

  if (dht11.read(pinDHT11, &temperature, &humidity, data)) {
    return;
```

```
  }
  for (int i = 0; i < 40; i++) {
    if (i > 0 && ((i + 1) % 4) == 0) {
    }
  }

  val = analogRead(water);
  H_in = analogRead(H);
  out = (int)((unsigned long)(H_in * 100) / 1024);

  lcd.setCursor(0, 0);
  lcd.print("temp = ");
  lcd.print((int)temperature);
  lcd.print(" C ");
  lcd.setCursor(0, 1);
  lcd.print("humidity= ");
  lcd.print((int)humidity);
  lcd.print("% =");
  lcd.print((unsigned)out);
  lcd.print(" ");
  delay(100);

  if ((val > 60) && (humidity == out) || humidity > out) {
    digitalWrite(relay, LOW);
  } else if ((val < 60) && (humidity != out)) {
    digitalWrite(relay, HIGH);
  }
}
```

# Program description :

1.  **Sensor Initialization:**

    -   **The program initializes the necessary components, including a Liquid Crystal Display (LCD), DHT11 sensor, analog humidity sensor, and a relay for the humidifier.**

2.  **Main Loop Operation:**

    -   **In a continuous loop, the program reads temperature and humidity values from the DHT11 sensor and calculates the humidity percentage from the analog humidity sensor.**

    -   **The obtained values are displayed on the LCD for real-time monitoring.**

3.  **Humidity Control:**

    -   **The program checks whether the current humidity level is below a pre-defined comfortable threshold ('desiredHumidityLevel').**

    -   **If the humidity falls below the threshold, the system activates the relay to turn on the humidifier for a fixed duration of 30 seconds.**

    -   **After the humidifier operates, the program rechecks the humidity level. If it is now within the desired range, the humidifier is turned off; otherwise, it continues to run.**

    -   **If the humidity is already within or above the desired range, the humidifier is turned off.**

4.  **User Feedback:**

    -   **The LCD provides continuous feedback, displaying the current temperature, humidity, and the status of the humidifier.**

# Implementation In the Laboratory