# Arduino Weather Station

## 1 Research

### 1.1 Problem Statement

There is a need for an affordable, customizable weather monitoring system for hobbyists, small-scale users, and educators. Existing solutions are often expensive or lack flexibility. Weather monitoring is crucial for farming, education, and disaster preparedness.

### 1.2 Why is it Important?

Weather monitoring is important for safety, disaster preparedness, and planning outdoor activities. It helps track environmental changes and promotes understanding and learning about climate and technology.

### 1.3 Background Research

Many Arduino-based weather stations have been created for DIY purposes. These projects often use components like temperature, humidity, pressure, and rain sensors, along with Arduino boards for data processing. Some advanced versions incorporate internet connectivity for real-time data logging.

### 1.4 Useful Components

**Sensors:**

DHT11 (Temperature and Humidity Sensor)
Analog sound sensor
Rain Gauge (to measure rainfall)
**Outputs:**
LCD display or OLED screen (to show weather data)
Bluetooth module
**Additional:**
Power supply (e.g., battery or solar panel for portability)
Enclosure for weatherproofing the components

### 1.5 Target Audience

**Hobbyists:** Electronics enthusiasts and students interested in learning about Arduino and weather data collection.

**Farmers and Gardeners:** Small-scale users who want to monitor local weather conditions for better crop management.

**Educators and Schools:** Teachers and students who can use the project as a learning tool for science and technology.

**Community Organizations:** Local groups tracking weather patterns for environmental or disaster preparedness purposes.

## 2 Development of Ideas

### 2.1 2.1 Idea 1:

Weather Monitoring Station Description: This design features a simple casing layout with an LCD on the front for display and sensors placed strategically. A DHT11 (temperature and humidity) sensor is placed outside for easy measurement. The power supply is routed neatly to ensure efficient functioning. The overall structure is functional and has ventilation which might lead to less overheating issues.

### 2.2 2.2 Idea 2:

Bluetooth Weather Station Description: This design is similar to the first design, contains the DHT 11 sensor for temperature and humidity data, and has a rain sensor to measure rain levels. Unlike the first model, which does not have an LCD to display the data, this model contains a Bluetooth module to send data to a mobile device. This design has more ventilation than the other design.

### 2.3 2.3 Idea 3:

Weather Monitoring Station (Final Design) Description: The third design contains the components DHT 11, a Rain sensor, an LCD and a Bluetooth module. This design combines the first two designs however removes the ventilation feature as it may destroy the modules inside the casing. This design uses both sensors to display data on the LCD and send data to a mobile device using a Bluetooth connection with the Bluetooth module.

### 2.4 Final Chosen Design

**Design 3** was selected because it provides both an LCD display and remote monitoring via Bluetooth. The structure is functional and durable, ensuring easy accessibility and accurate data collection.

# 3 Design Plan

## 3.1 Materials List

Arduino Uno
- DHT11 Temperature and Humidity Sensor
- Rain Sensor
- LCD Display
- Bluetooth Module
- Power Supply
- 3D-printed enclosure

## 3.2 Flowchart for Code

Flowchart Link: View Here

# 4 Testing and Evaluation

## 4.1 Initial Testing

The first test was conducted indoors to verify sensor readings. The temperature and humidity values were accurate, but the rain sensor required calibration. The LCD display was difficult to read due to excessive text displayed simultaneously.

## 4.2 Feedback & Adjustments

Improved display clarity by implementing a screen-switching feature every 5 seconds.
- Adjusted LCD contrast and text size for better readability.
- Calibrated the rain sensor for more accurate readings.
- Considered adding an audio or LED alert for rain detection.

# 5 Iterations

## 5.1 Changes Made

**3D Design Adjustments:** Added space to fit the DHT11 sensor properly and increased the overall size of the casing for better airflow.

**Buzzer Removal:** Initially attempted to integrate a buzzer for rain alerts but removed it due to constant buzzing, which was not user-friendly.

**Rain sensor and Bluetooth removal:** In the design i chose to make there was a Bluetooth module and rain sensor. After testing the circuit and getting feedback i realized that adding a rain sensor would mean that there would be a a lot of power usage and also means that the weather box would be placed somewhere that rain can be sensed from the sensor that means that there is risk of damaging the components inside the 3d casing. I removed the Bluetooth

module because of the external power usage and also because it would not be needed because of the LCD.

**LCD Data Display Improvement:** Implemented screen-switching functionality to cycle through the temperature, humidity, and rain data instead of showing all the information at once.

# 6 Final Reflection

The Arduino Weather Station, built using an Arduino Uno, a DHT11 sensor, and a LCD, is a user-friendly and cost-effective alternative to commercial weather monitoring systems. Although the project successfully displays temperature and humidity, there are areas for improvement.

## 6.1 Challenges & Lessons Learned

**Casing Design:** Learned how to design and modify a 3D printed case to efficiently accommodate components.

**Sensor Integration:** Gained experience coding and integrating multiple sensors for a functional system.

**Future Improvements:** Make the outer casing more resistant to freezing temperatures and water damage, and add additional sensors such as a wind speed sensor. Another improvement would be making a way for the Arduino board to have more power to have a Bluetooth module.

# 7 Final Code

Listing 1: Arduino Code

```
#include <LiquidCrystal_I2C.h>
#include <dht11.h>
#define DHT11PIN A0

dht11 DHT11;
LiquidCrystal_I2C lcd(0x27, 16, 2);

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

  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("Weather Box");
  delay(1000);
  lcd.clear();
```

```
}

void loop() {

  int chk = DHT11.read(DHT11PIN);

  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11.humidity, 2);

  Serial.print("Temperature  (C): ");
  Serial.println((float)DHT11.temperature, 2);

  lcd.setCursor(1, 0);
  lcd.print("Humidity(%):");
  lcd.setCursor(1,1);
  lcd.print((float)DHT11.humidity, 2);

  delay(2000);
  lcd.clear();

  lcd.setCursor(1, 0);
  lcd.print("Temperature(C):");
  lcd.setCursor(1,1);
  lcd.print((float)DHT11.temperature, 2);

  delay(2000);
  lcd.clear();
}
```