First try:

To create an **Automatic Plant Watering System** using **Arduino**, you can integrate several sensors like the **Soil Moisture Sensor**, **DHT11 (Temperature & Humidity Sensor)**, and a **Rain Sensor** to make it more versatile. You'll also need an **LCD display** to show the readings. When the soil moisture is below a certain threshold, the system will activate a water pump to water the plant. Below is the code to help you implement this project.

**Components:**

1. **Arduino Uno**
2. **Soil Moisture Sensor**
3. **DHT11 Sensor**
4. **Rain Sensor**
5. **16x2 LCD Display (I2C or Parallel)**
6. **Water Pump (with Relay Module)**
7. **Jumper Wires**
8. **Power Supply**
9. **Breadboard**

**Wiring Connections:**

**1. Soil Moisture Sensor:**

* **VCC**: Arduino **5V**
* **GND**: Arduino **GND**
* **Analog Output (A0)**: Arduino **Analog Pin A0**

**2. DHT11 Sensor:**

* **VCC**: Arduino **5V**
* **GND**: Arduino **GND**
* **Data Pin**: Arduino **Digital Pin 2**

**3. Rain Sensor:**

* **VCC**: Arduino **5V**
* **GND**: Arduino **GND**
* **Analog Output (A0)**: Arduino **Analog Pin A1**

**4. LCD Display (I2C):**

* **SDA**: Arduino **A4**
* **SCL**: Arduino **A5**

**5. Water Pump (Relay Module):**

* **VCC**: Arduino **5V**
* **GND**: Arduino **GND**
* **Input (IN)**: Arduino **Digital Pin 7**

**Arduino Code (C++):**

cpp

Copy code

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

// Define the pins for the sensors

#define DHTPIN 2 // Pin connected to the DHT11 sensor

#define DHTTYPE DHT11

#define soilMoisturePin A0 // Soil moisture sensor pin

#define rainSensorPin A1 // Rain sensor pin

#define pumpPin 7 // Relay connected to water pump

DHT dht(DHTPIN, DHTTYPE);

// Initialize the LCD (for I2C LCD)

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Address 0x27 for a 16x2 LCD

// Soil moisture threshold (adjust as needed)

int moistureThreshold = 500; // Value to decide when to water the plant

void setup() {

// Initialize the LCD

lcd.begin();

lcd.backlight();

// Initialize sensors

dht.begin();

pinMode(soilMoisturePin, INPUT);

pinMode(rainSensorPin, INPUT);

pinMode(pumpPin, OUTPUT);

// Start with the pump turned off

digitalWrite(pumpPin, LOW);

}

void loop() {

// Read sensor values

int soilMoistureValue = analogRead(soilMoisturePin);

int rainValue = analogRead(rainSensorPin);

float temp = dht.readTemperature(); // Reading temperature in Celsius

float humidity = dht.readHumidity(); // Reading humidity

// Print sensor values to the LCD

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.print(temp);

lcd.print("C ");

lcd.print("Humi: ");

lcd.print(humidity);

lcd.setCursor(0, 1);

lcd.print("Soil: ");

lcd.print(soilMoistureValue);

lcd.print(" Rain: ");

lcd.print(rainValue);

// Condition for watering

if (soilMoistureValue < moistureThreshold && rainValue > 300) {

// Soil is dry and no rain, water the plant

digitalWrite(pumpPin, HIGH); // Turn on the water pump

lcd.setCursor(0, 1);

lcd.print("Watering Plants ");

} else {

// Soil is moist or it's raining, don't water the plant

digitalWrite(pumpPin, LOW); // Turn off the water pump

lcd.setCursor(0, 1);

lcd.print("No Water Needed ");

}

// Wait 2 seconds before the next loop

delay(2000);

}

**Code Breakdown:**

1. **DHT Sensor:**
   * Reads the temperature and humidity values using the DHT11 sensor.
2. **Soil Moisture Sensor:**
   * Reads the soil moisture level from the analog pin. If it falls below a certain threshold, the water pump is triggered.
3. **Rain Sensor:**
   * The system checks for rain. If rain is detected (value greater than a certain threshold), the water pump does not activate, even if the soil is dry.
4. **LCD Display:**
   * Displays the current temperature, humidity, soil moisture, and rain sensor readings.
5. **Water Pump:**
   * Connected through a relay module. It turns on when the soil is dry and there is no rain detected.

**How It Works:**

* The **Soil Moisture Sensor** detects whether the soil is dry or not.
* The **Rain Sensor** ensures the pump doesn't activate if it detects rain.
* The **DHT11 sensor** measures the environment’s temperature and humidity, and this data is displayed on the LCD.
* If the soil moisture is below the threshold and no rain is detected, the relay activates the water pump to water the plant.

This project can be expanded with more features like logging data to an SD card or adding wireless communication. Let me know if you need help with further customization!

Your code looks mostly correct, but there are a few improvements and optimizations we can make to ensure efficient execution on an Arduino Uno:

**Corrections and Optimizations:**

1. **LCD Initialization**: The lcd.begin() should include the number of columns and rows.
2. **Sensor Read Handling**: Add error handling in case the DHT sensor fails to read values.
3. **Efficient LCD Display**: Clear only the necessary parts of the LCD to avoid flickering and keep important messages visible.
4. **Rain Sensor Value Handling**: Ensure that the value for rain detection is properly interpreted.

**Optimized and Corrected Code:**

cpp

Copy code

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

// Define the pins for the sensors

#define DHTPIN 2 // Pin connected to the DHT11 sensor

#define DHTTYPE DHT11 // DHT11 sensor type

#define soilMoisturePin A0 // Soil moisture sensor pin

#define rainSensorPin A1 // Rain sensor pin

#define pumpPin 7 // Relay connected to water pump

DHT dht(DHTPIN, DHTTYPE);

// Initialize the LCD (for I2C LCD)

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Address 0x27 for a 16x2 LCD

// Soil moisture threshold (adjust as needed)

int moistureThreshold = 500; // Value to decide when to water the plant

void setup() {

// Initialize the LCD with 16 columns and 2 rows

lcd.begin(16, 2);

lcd.backlight(); // Turn on the backlight

// Initialize sensors

dht.begin();

pinMode(soilMoisturePin, INPUT);

pinMode(rainSensorPin, INPUT);

pinMode(pumpPin, OUTPUT);

// Start with the pump turned off

digitalWrite(pumpPin, LOW);

// Initialize Serial for debugging (optional)

Serial.begin(9600);

}

void loop() {

// Read sensor values

int soilMoistureValue = analogRead(soilMoisturePin);

int rainValue = analogRead(rainSensorPin);

// Read temperature and humidity from the DHT sensor

float temp = dht.readTemperature(); // Reading temperature in Celsius

float humidity = dht.readHumidity(); // Reading humidity

// Check if DHT sensor reading is valid

if (isnan(temp) || isnan(humidity)) {

lcd.setCursor(0, 0);

lcd.print("DHT Error! ");

Serial.println("Failed to read from DHT sensor!");

delay(2000);

return;

}

// Print sensor values to the Serial Monitor (for debugging)

Serial.print("Soil: "); Serial.print(soilMoistureValue);

Serial.print(" | Rain: "); Serial.print(rainValue);

Serial.print(" | Temp: "); Serial.print(temp);

Serial.print("C | Humidity: "); Serial.println(humidity);

// Update LCD display

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.print(temp);

lcd.print("C H:");

lcd.print(humidity);

lcd.print("% ");

lcd.setCursor(0, 1);

lcd.print("Soil: ");

lcd.print(soilMoistureValue);

lcd.print(" Rain: ");

lcd.print(rainValue);

// Watering logic: check moisture level and rain status

if (soilMoistureValue < moistureThreshold && rainValue > 300) {

digitalWrite(pumpPin, HIGH); // Turn on the water pump

lcd.setCursor(0, 1);

lcd.print("Watering Plants ");

Serial.println("Watering Plants");

} else {

digitalWrite(pumpPin, LOW); // Turn off the water pump

lcd.setCursor(0, 1);

lcd.print("No Water Needed ");

Serial.println("No Water Needed");

}

// Wait 2 seconds before the next loop

delay(2000);

}

**Key Improvements:**

1. **LCD Initialization**:
   * The lcd.begin(16, 2) is correctly used to define a 16x2 LCD.
2. **DHT Error Handling**:
   * We added a check for the temperature and humidity readings. If there is an error reading from the DHT sensor, the program prints an error message on the LCD and skips that loop iteration.
3. **Efficient LCD Updates**:
   * The code clears or updates only necessary parts of the LCD. This avoids unnecessary flickering caused by clearing the entire screen.
4. **Serial Monitoring (Optional)**:
   * For debugging, the code prints values to the Serial Monitor, which can be viewed in the Arduino IDE by clicking **Tools** > **Serial Monitor**.
5. **Rain Sensor Value**:
   * The condition rainValue > 300 assumes that a higher value means no rain (adjust the threshold based on your sensor).

**Notes:**

* **Threshold Adjustments**: You may need to adjust the threshold values for the **soil moisture** and **rain sensor** based on your specific sensor calibration.
* **Power Considerations**: Ensure your Arduino has enough power to run the sensors, LCD, and pump.

This should now run efficiently on your Arduino Uno with the LCD, DHT sensor, soil moisture sensor, and rain sensor integrated. Let me know if it works or if you encounter any other issues!

**Rain sensor code:**

/\*

\* How to use rain sensor.

\* Created by SriTu Tech team.

\* Read the code below and use it for any of your creations.

\*/

void setup() {

Serial.begin(9600); // enable serial monitor

pinMode(4, OUTPUT); // define LED pin

}

void loop() {

int value = analogRead(A3); // read value from the sensor

Serial.print("Value = ");

Serial.println(value); // print the sensor value to the serial monitor

if (value < 300) { // check condition for heavy rain

digitalWrite(4, HIGH); // turn LED on

Serial.print("Heavy rain LED on "); // print message to serial monitor

} else {

digitalWrite(4, LOW); // turn LED off

}

}

**Temperature sensor:**

#include "DHT.h"

#define DHTPIN 2 // Digital pin connected to the DHT sensor

// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13, or 14

// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whichever DHT type you are using!

// #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

#define DHTTYPE DHT11 // DHT 11

// #define DHTTYPE DHT21 // DHT 21 (AM2301)

Serial.println(F("DHTxx test!"));

dht.begin();

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahrenheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

Serial.print(h);

Serial.print(F("% Temperature: "));

Serial.print(t);

Serial.print(F("°C "));

Serial.print(f);

Serial.print(F("°F Heat index: "));

Serial.print(hic);

Serial.print(F("°C "));

Serial.print(hif);

Serial.print(F("°F"));

**Soil sensor code:**

// Welcome

// Electronics University

int relayPin = 8;

int sensor\_pin = A0; // Soil Sensor input at Analog PIN A0

int output\_value;

void setup() {

Serial.begin(9600);

pinMode(relayPin, OUTPUT);

pinMode(sensor\_pin, INPUT);

Serial.println("Reading From the Sensor ...");

delay(2000);

}

void loop() {

output\_value = analogRead(sensor\_pin);

output\_value = map(output\_value, 550, 10, 0, 100);

Serial.print("Moisture : ");

Serial.print(output\_value);

Serial.println("%");

if(output\_value < 0) {

digitalWrite(relayPin, LOW);

} else {

digitalWrite(relayPin, HIGH);

}

delay(1000);

}