SECTION 5. PROBLEM OF INNOVATIVE EDUCATION AND DECISION MAKING

DOI https://doi.org/10.30525/978-9934-26-597-6-36

DETERMINING PLANT STRESS LEVELS USING ARDUINO

Vadim Obukhov*, Asadillo Abdullayev
Fergana state technical university, Ferghana, 150100, Uzbekistan
**Corresponding author's e-mail: wendigo_chelsea@mail.ru*

Abstract
Plant stress is the main factor affecting the yield and general health of plants. Stressors, such as drought, salinity, extreme temperatures and pathogenic attacks, destroy cell processes, which leads to morphological and physiological changes in plants. Early detection of stress is crucial for mitigating its impact on agriculture. Various methods for plant stress determination, including physiological indicators, molecular markers, and remote sensing technologies, have been developed to monitor stress levels. These techniques provide valuable insights into plant responses and resilience. This review explores the advancements in plant stress detection, focusing on the integration of traditional and modern approaches to improve crop management and ensure food security.
Keywords: plant stress, drought, salinity, remote sensing, productivity

1 Introduction
The primary microclimate, namely humidity and temperature, serve as the most important criteria for survival. Nevertheless, with more attention to temperature diets, inexperienced crops often make mistakes without worrying about humidity, believing that watering is sufficient. And they fight the problems and are surprised. They focus on not reaching harvest by "eating" with plants. On the other hand, a knowledge of the type of humidity a plant requires, and a way to significantly improve the appearance of the green, literally refer to it to life, and push it forward into a positive harmonious development, rich and fruitful colour. The lack of humidity in the air indicates a fragility at the edge of the leaf plate, yellow, its falling, falling, falling, wrinkle. Stop and the flowers are delicate, fall again or completely attached.

Representatives of tropical and equatorial latitude flowers are particularly sensitive to reduced humidity. For example, leaf leaves lower water to 50-30% beyond characteristic disease. But first, excessive humidity is dangerous for the high risk of disintegration, mold, fungal diseases and harmful reproductive insects. Sukulants are very sensitive to high humidity, but can easily adapt to "the environment".

2 Materials and methods

With an Arduino controller and inexpensive sensor, you can keep your internal plants in good condition by monitoring parameters such as humidity, heat, and light levels. In the simplest performance version, the plant control system determines the temperature brightness that depends on the light level, soil humidity, and plant condition. The device reads these readings on a USB and is displayed on the computer screen. Start your project with the simplest steps, use LED and LCD screens to make it more complex, and learn more about Arduino and other electronic components. All electronic components for three different executions are available on the Shack Made radio designer: Ultimate Microcontroller Pack. Components can also be purchased separately [1].

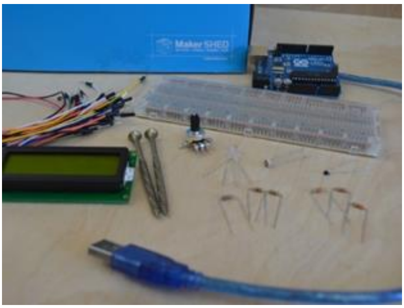

Figure 1. Getting started: measuring light, heat and humidity levels

The first step requires an Arduino UNO microcontroller, a flipper, three resistors rated on 10 KOHM, an installation guide, a terrist (temperature sensor), a photorewant (light sensor), a screw and a steel nut. You also need to create two separators on the soil humidity sensor (on the screw). The plastic has two holes in diameter to suit the screw size. Create a second separator, bring the screws into two separators, and place a short distance from each other. I definitely determined the soil sensor. This is done using an acrylic polymer laser cut in 1/16 size. It acts as a separator between small segments of 1/16 inch diameter stainless steel stainless steel sticks and is commonly used to weld argon arcs [5]. The following parts models that need to be cut into acrylic polymers that can be loaded.

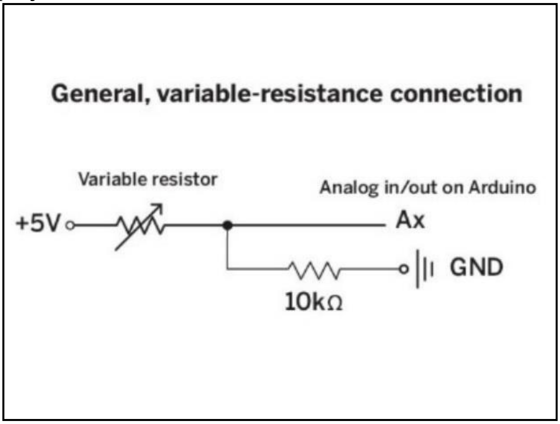

Figure 2. General, variable-resistance connection

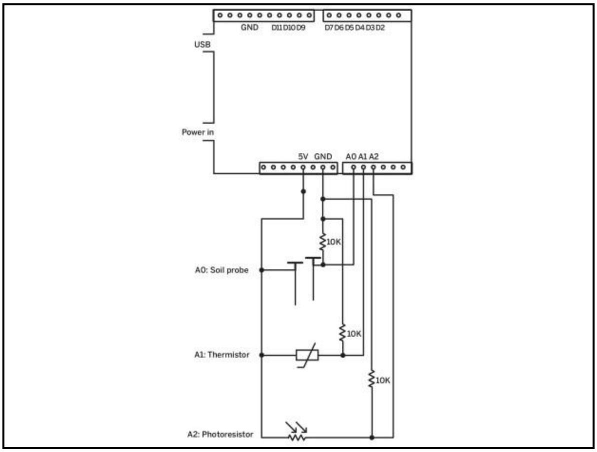

Figure 3. Connection scheme

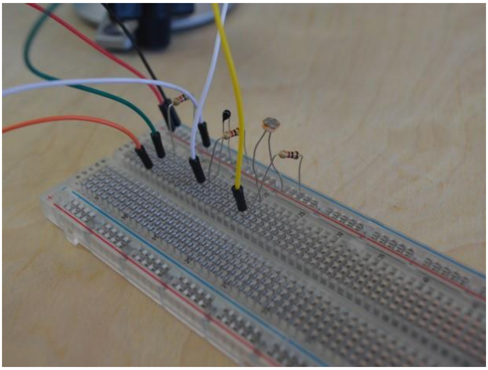

Figure 4. Connection scheme

The operating principle of the three sensors used is the same. They provide a change in electrical resistance depending on the amount of moisture in the soil, or the level of heat or light that can be detected. A simple circuit can usually be used to read the resistance measurements of the variable alduino as shown in the diagram above. Here we use this layout for three sensors. This reveals humidity, temperature and level of soil lighting and creates a chain, connecting them together, as shown in the second diagram. In accordance with the electric chain, collect the chain on the layout, as indicated in the third diagram. The operator thermistor and the photographs are located on the location; The soil moisture sensor is 2 long strands, each wound on the screw and fixed with a nut. The soil moisture sensor has an analog output microcontroller Arduino, the a1 departure terminator and the photoresistor connected to the output A2 [8].

Programming the Arduino

```
#include <DHT.h>
// Determining the sensor pins
#define DHT_PIN 2 // Pin for temperature and humidity sensor
#define SOIL_PIN A0 // Pin for soil moisture sensor
#define LDR_PIN A1 // Pin for photoresistor
// Set the DHT sensor type
#define DHT_TYPE DHT11 // Use DHT22 if you use it
DHT dht(DHT_PIN, DHT_TYPE);
void setup() {
```

```
Serial.begin(9600); // Initializing the serial port
dht.begin(); // Initializing the temperature and humidity sensor
}
void loop() {
// Reading data from the soil moisture sensor
int soilMoisture = analogRead(SOIL_PIN);
// Reading data from the temperature and humidity sensor
float humidity = dht.readHumidity();
float temperature = dht.readTemperature();
// Reading data from the photoresistor
int lightLevel = analogRead(LDR_PIN);
// Output data to the serial monitor
Serial.print("Soil Moisture: ");
Serial.print(soilMoisture);
Serial.print(" | Temperature: ");
Serial.print(temperature);
Serial.print(" C | Humidity: ");
Serial.print(humidity);
Serial.print(" % | Light Level: ");
Serial.println(lightLevel);
// Watering, temperature and light conditions
if (soilMoisture < 400) {
Serial.println("Soil is dry. Consider watering the plant.");
} else if (soilMoisture > 600) {
Serial.println("Soil is wet. No need to water.");
}
if (temperature > 30) {
Serial.println("It's too hot! Consider moving the plant to a cooler place.");
} else if (temperature < 15) {
Serial.println("It's too cold! Consider moving the plant to a warmer place.");
}
if (lightLevel < 200) {
Serial.println("Light level is low. Consider moving the plant to a brighter spot.");
}
delay(2000); // Delay of 2 seconds before the next measurement
}
```

As a rule, use the following code to read the signs of a change in resistance of the sensor connected to the Anduino analog pins:

sensorValue = analogread(inputPin);

Sensor Value is a sensor indicator, and the input is the number of analog inputs to which the sensor is connected. To read the three sensors, repeat this code to the three sensors, add a variable description, and create a serial port configuration. Consequently, we have the following code, which you must open in Arduino Ide, and download it to the Arduino microcontroller [2].

Sensor placement and calibration

Insert a humidity level sensor into the ground next to the factory, place a list of photos that will provide the same amount of light as the factory sheet, and adjust the thermistor in the photo list. Provides power to your Arduino, opens the computer's coherent monitor and displays sensor readings that are updated every second. To be very careful with the internal plant, the sensor must be adjusted to all the plant conditions. An extended sensor can be used to calibrate the sensor according to its adaptation. Compared to the Arduino dimensions from the welded stem home humidity sensor, you can find below the signs of an industrial humidity sensor. Inside: Two sensors are displayed. Very dry soil: 2% of vegetables, 5% of the home sensor. Small humidity: 7%, 150% – home sensor. Soft moisture, average cost: 8%, 250% – Home Sensor. Moist Soil: 28%, 370% – Home Sensor. Very humid soil: 51%, 385% – Home sensor. Painted with a water sensor: 85%, 480% -H -Sensor. Therefore, if you want to reach around 28% of the soil humidity, you should observe how the sensor showed 385 and use this value as a threshold [6].

Now you can add an LED to show the plant's condition in the diagram. I used blue to indicate the need for water and water, and green was shown with red light and temperature. All conditions are displayed by the RGB LED. (Of course, there are a variety of options available!). Attach the RGB LED and three resistors to the 330th nominal ohm. Depending on the scheme shown in the first diagram, the LED can be connected to the Arduino microcontroller. In the diagram, the LED is labeled as a delta with a common connection. With common LEDs, the longest pin should be connected to ground (if the connection is wrong, if the LED can be extended!) [3]. The resistor regulates the LED current. If the current is very high, the LED can come out in seconds rather than 10,000 hours later [7].

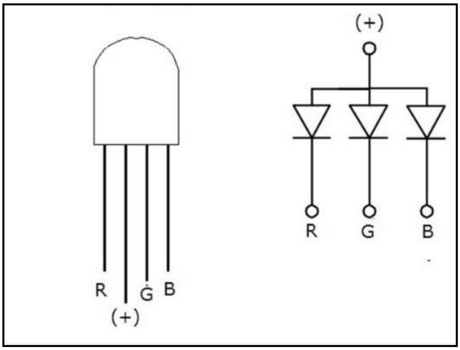

Figure 5. Adding RGB LEDs

Connecting the LED to the circuitry

With RGB LEDs, red, green, and blue elements can have a common cathode or a common anode. This is a 4-output LED (please check the technical sheet if you are using a specific RGB LED with a common anode or a common cathode). Designer Designer Designer: The LED LEDs in the Ultimate Microcontroller Pack have a common anode. Therefore, connect as shown on the schematic. Depending on the illustration, connect the LED to the + 5V output and to the digital inputs D9, D10 and D1 of the Arduino microcontroller.

Reprogramming the Arduino

```
// Initializing variables for sensor pins
int moistPin = 0;
int tempPin = 1;
int lightPin = 2;
// Initialization of variables for storing sensor data
int moistVal = 0;
int tempVal = 0;
int lightVal = 0;
// Initialization of variables for LEDs
int redPin = 11;
int greenPin = 10;
int bluePin = 9;
void setup() {
// Initializing the serial port
Serial.begin(9600);
```

```
// Setting the LED pins to output mode
pinMode(redPin, OUTPUT);
pinMode(greenPin, OUTPUT);
pinMode(bluePin, OUTPUT);
// Disabling all LEDs
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, HIGH);
digitalWrite(bluePin, HIGH);
}
void loop() {
// Here should be the code for working with sensors and LEDs
}
```

The main advantage of this project is the function call. Instead of always printing the same code, you simply convert it into a function and call this function. For example, the following code turns off the three LED colors [9]:

```
digitalWrite(redPin, HIGH – red
lead, High potential);
digitalWrite(greenPin, HIGH – green lead, High potential);
digitalWrite(bluePin, HIGH – blue
lead, High potential);
```

Instead of entering this code several times to turnoff the LED, I convert this code into a function.

```
void off()
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, HIGH);
digitalWrite(bluePin, HIGH);
}
```

You can now run this code if you wish, then off Function (); You can create more complex things because you know the basics of programming well. Therefore, we have the following code – in the pppled.ino file. The code will lead to the fact that the LED will be lit in one of the colors for 5 seconds, according to the states of your plant: the LED will be lit in blue for 1 second, if the soil is wet. The LED will illuminate the red if the plants are hot. When plants are exposed to the sun, the LEDs are illuminated with green.

Complicating the project: Displaying measurement readings on the LCD display

Next, you add LCD-Display text to your design and cycle with the withdrawal of the measurements of three sensors, each reader being displayed within 4 seconds (LEDs operate according to previously established conditions). Connect the 16-grain LCD screen from the 10th point with a pentimo metre to adjust the manufacturer's designer brightness, the Ultimate Microcontroller Pack. The conductors must be soldered to the LCD –display [10].

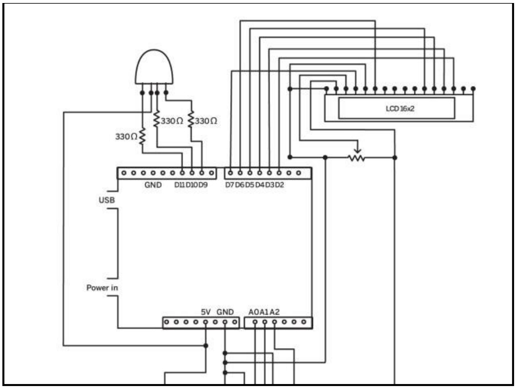

Figure 6. Connect the LCD display and verify operation

LCD-display is slightly abnormally connected. Don't worry, we will provide you with all the information you need to fully understand this project! For those interested, follow the diagram and connect the LCD-DICE as the following tenor function: LCD impermeasure, conclusion N°1, connected to the earth in this place. The LCD-display, withdrawn n°2, is connected to an external potentiometer with five external outputs, but this is not a problem. You can choose. LCD-display, output n°3 is connected to the throat spindle of the potentiometer, which is the central output. LCD -display, output number 4 is connected to the Arduino. Digital Conclusion n°7 – This is different from the factory library of LCD-displays. LCD-display, output n°5 is connected to the Earth's potentiometer and external output. LCD-display, Brooch n°6, connected to Arduino, digital output #6, which is different from the factory library for LCD screens. LCD -DICE, pin N°7-#10 is not connected. LCD-Display, PIN#11 is connected to the Arduino, digital output #5-Display, PIN#12 is connected to the Arduino, Digital Output #4 LCD-Display. Pins #14 and #15. The above Arduino educational materials provide all the information you need to control the LCD screen [4].

3 Results

Reprogramming the Arduino

// Potted plant protector with RGB indicator LED and LCD
// Connecting the library for working with LCD

```
#include <LiquidCrystal.h>
// Initializing variables for sensor pins
int moistPin = 0;
int tempPin = 1;
int lightPin = 2;
// Initialization of variables for storing sensor data
int moistVal = 0;
int tempVal = 0;
int lightVal = 0;
// Initialization of variables for LEDs
int redPin = 11;
int greenPin = 10;
int bluePin = 9;
// Initialization of LCD with interface pin numbers
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
```

After connecting the LCD screen, you can fully refer to the program code for the internal factor protector. If you are satisfied with the LCD screen, you can download the code (pppelbuild.ino) and write it to the Arduino microcontroller.

The project is ready! Now your plants will say exactly what they want. Here are some additional components that are suitable to improve this project: Control Foundation: Add a wool with relays to control water pumps, radiators or light sources. "The Arduino library can be used for tasks such as light sources, so sunlight + 16 hours of light is added. Food Monitoring: Make the Arduino microcontroller fall asleep, wake up and display measurements every few minutes. This will considerably increase the life of the battery. This will also allow you to extend the lifespan of your soil humidity sensor before you start correcting. Solar energy source: add solar panels and batteries for autonomous power for the defender. Connection tools: Share your data with others and organize remote access to Ethernet, Wi-Fi or Shields help for your mobile organization. Improved Sensors: Research many other sensor options, such as PH, optical spectrum, humidity, and more to internalize factory defenders to the next level of technology.

4 Discussion

When plants encounter unfavorable factors, they can feel stressed. Stress is the biological resistance response of plants to factors that negatively affect your health. And plants can be repulsed. They developed a lot of protective mechanisms, including pathogens, drought, fever, cold and herbivores. These defense mechanisms are the reason plants survive. As a grower, you can help your crop fight stressors, promoting full bloom and high productivity. In this article, we will discuss seven stressors.

Stress factor 1

Absent or excessive water. The stress caused by these factors manifests itself in a Wilter. It is very important to know exactly how much your plant needs, because each variety has different needs. In our article, “How much water do plants need?”

Stress factor 2

Food Habitat. All crops do not have just the right habitat. Each variety has its own unique needs. Thus, plants need a lot of light and other plants need high temperatures in the room. Discover these characteristics in advance and quickly place your plants in the desired environment. For this, how do you understand that your harvest is developing in a disadvantaged environment? If brown or black spots appear on the leaves, this means that this habitat is too hot or too dry. Do plants have fallen leaves? After that, the room temperature is too low or the project is in the factory. For additional protection, a wooden board can be placed under the pot. Are plants yellow to plants? It also indicates that the medium temperature is too low. Are plant stems free? So you're too dark in the pot. Stress can also be caused by sudden movement of plants in an outdoor room. Initially, plants should be placed in shelters so that the transition is not so clear.

Stress factor 3

Food shortages. When a plant\'s leaves change color, it usually indicates a lack of nutrients. Trace elements found in fertilizers are responsible for the growth of the plant and the operation of certain functions. If it's too little, or conversely, too many trace elements can harm the plant. Learn more in this article, "What are the elements of tracing?" If you have taken all the necessary measures, and the color of your leaves continues to change, the problem is probably your PH or level of acidity. To configure the PH value, you can use pH and pH+ products from the range.

Stress factor 4

The air is too dry. In winter, the air is a dryer, making the leaves of the plant dry and dusty. If this happens, the plant will not receive enough sun. Therefore, we recommend dust regularly wiping plants, for example, with a damp cloth.

Stress factor 5

The air humidity is too high. If the humidity is high, the plant cannot evaporate the moisture. This is harmful, because then in the roots of the plant, there is not enough space to absorb new nutrients. Therefore, we recommend that you buy an extractor, a fan or a humidifier.

Stress factor 6

Transplant into another pot. For plants, transplanting another pot is a major change. The plant loses its roots, is in a new environment and has to get used to the new soil. Therefore, its leaves can become yellow or fall. If the soil mixture in the saucepan contains the necessary nutrients, the plant should be in order. It usually takes a while for it to be completely used to the new environment and continues to grow.

Stress factor 7

Diseases and parasites. Plants may also encounter diseases or parasites. These include, for example, pathogenic fungi, root rot and outbreaks of pest breeding insects. In the latter case, it is advisable to move affected plants away from other plants.

5 Conclusions

Resistance to major biological and abiotic stress is one of the main requirements presented to modern crops and their types of cultural technology. To achieve stable results under a variety of environmental conditions, it is important not only to select the right variety, but also to apply cultural practices that can maximize the potential defenses of the plant. In many crops, the problem of long-term integrated resistance to organic and non-biotic stresses has not yet been resolved, and plant protection chemicals must be used to obtain satisfactory yields. This problem is particularly sharp for tomatoes, which are widely used in foods for children and adults. Therefore, the use of chemical defense against tomato disease should be limited. At the same time, in Veraros, a certain number of tomato diseases in the context of the degradation of plant hospitality can cause deaths of up to 40 ÷ 60% in a few years, and deaths of up to 80% of the yield. Furthermore, the loss of differences under weather conditions and other abiotic factors is important. All this is based on the priorities of agriculture with an adaptive landscape, seeking the most environmental methods and encouraging measures to reduce and stabilize plant productivity [11].

References

[1] Zhu, C., Ding, Y., & Liu, H. (2011). MiR398 and plant stress responses. Physiologia plantarum, 143(1), 1-9.

[2] Mittler, R., Zandalinas, S. I., Fichman, Y., & Van Breusegem, F. (2022). Reactive oxygen species signalling in plant stress responses. Nature reviews Molecular cell biology, 23(10), 663-679.

[3] Maestre, F. T., Valladares, F., & Reynolds, J. F. (2005). Is the change of plant–plant interactions with abiotic stress predictable? A meta-analysis of field results in arid environments. Journal of Ecology, 93(4), 748-757.

[4] Gaspar, T., Franck, T., Bisbis, B., Kevers, C., Jouve, L., Hausman, J. F., & Dommes, J. (2002). Concepts in plant stress physiology. Application to plant tissue cultures. Plant growth regulation, 37, 263-285.

[5] Maestre, F. T., Valladares, F., & Reynolds, J. F. (2006). The stress-gradient hypothesis does not fit all relationships between plant–plant interactions and abiotic stress: further insights from arid environments. Journal of Ecology, 94(1), 17-22.

[6] Hsiao, T. C. (1973). Plant responses to water stress. Annual review of plant physiology, 24(1), 519-570.

[7] Hsiao, T. C. (1973). Plant responses to water stress. Annual review of plant physiology, 24(1), 519-570.

[8] Wang, M., Zheng, Q., Shen, Q., & Guo, S. (2013). The critical role of potassium in plant stress response. International journal of molecular sciences, 14(4), 7370-7390.

[9] Baret, F., Houlès, V., & Guerif, M. (2007). Quantification of plant stress using remote sensing observations and crop models: the case of nitrogen management. Journal of experimental botany, 58(4), 869-880.

[10] Ghatak, A., Chaturvedi, P., & Weckwerth, W. (2018). Metabolomics in plant stress physiology. Plant genetics and molecular biology, 187-236.

[11] Behmann, J., Steinrücken, J., & Plümer, L. (2014). Detection of early plant stress responses in hyperspectral images. ISPRS Journal of Photogrammetry and Remote Sensing, 93, 98-111.

Authors

Vadim Obukhov, 31.01.1991, Ferghana, Uzbekistan

Assistant of Department of Information systems and technologies, Faculty of Information technology and telecommunications, Fergana state technical university.

Information technology, telecommunications, networks and servers.

Asadillo Abdullayev, 18.04.2003, Andijan, Uzbekista

4th year student of Fergana State Technical University

Information technology, telecommunications, networks and servers.