Green Revolution: Boosting Plant Performance

(research paper on photosynthesis in C3, C4 and CAM plants)

By **Bodrug Denis**

**denis.bodrug@ungheni.edu.md**

**Abstract:**

*This research aims to explore more efficient photosynthesis processes and identify their potential applications for improving various aspects of life.*

**Thesis :**

With a growing global population and depleting resources, enhancing crop yields and resource efficiency is critical. Incorporating C4 photosynthesis, known for its efficacy, into C3 crops holds promise. C4 plants excel in hot climates, making them attractive for boosting productivity. Despite evolutionary transitions from C3 to C4, engineering faces challenges, requiring a deep understanding of gene regulation and anatomical complexities. Previous attempts lacked systemic comprehension, hindering success. This research aims to unravel mechanisms governing Kranz anatomy and cell-specific expression in C3 and C4 plants, crucial for successful engineering. The paper reviews past efforts, recognizing limitations, and advocates for a systemic approach with recent innovations to overcome challenges.

**Research plants:**

The research will focus on three plant species: C3 grass, C4 corn, and CAM aloe vera.

**Equipment Used:**

* PASCO Sensors: Temperature, Light, CO2, O2
* Custom Arduino Installations: Temperature Control, Airflow Control.

**Use of equipment:**

*The utilization of a diverse array of sensors serves a pivotal role in our research to ensure comprehensive and accurate data collection. Each sensor plays a specific role in monitoring key environmental factors that directly influence the photosynthetic processes of the studied plants.*

Temperature sensors track variations in temperature, influencing enzymatic reactions critical for photosynthesis. Light sensors gauge the intensity of light, a fundamental factor in the light-dependent reactions of photosynthesis. CO2 and O2 sensors monitor the levels of these gases, directly impacting carbon fixation and the overall efficiency of photosynthesis.

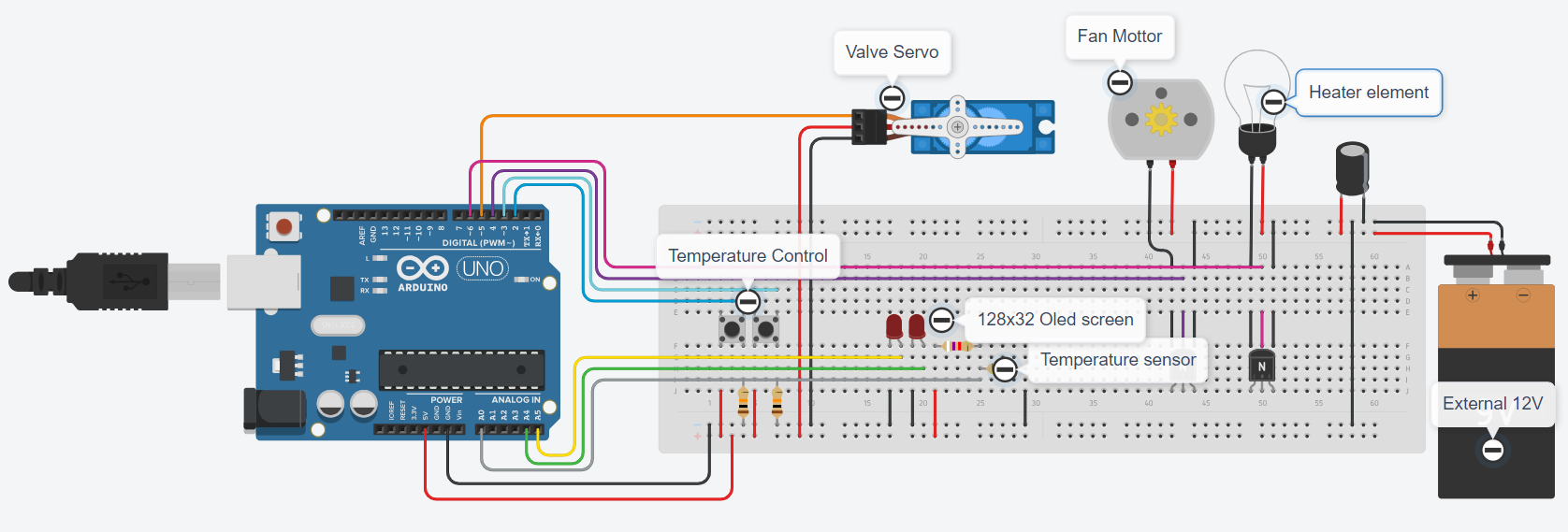
*Incorporating these sensors from PASCO ensures precision and reliability in data collection. Additionally, the custom-built Arduino installations, equipped with temperature control, airflow control, and data logging capabilities, enhance the experimental setup's control and monitoring capabilities.*

*Collectively, the use of these sensors enables a thorough examination of the intricate interplay between environmental variables and plant photosynthetic efficiency, contributing to a more nuanced and accurate analysis of our research findings.*

**To repeat the experiment and data readings:**

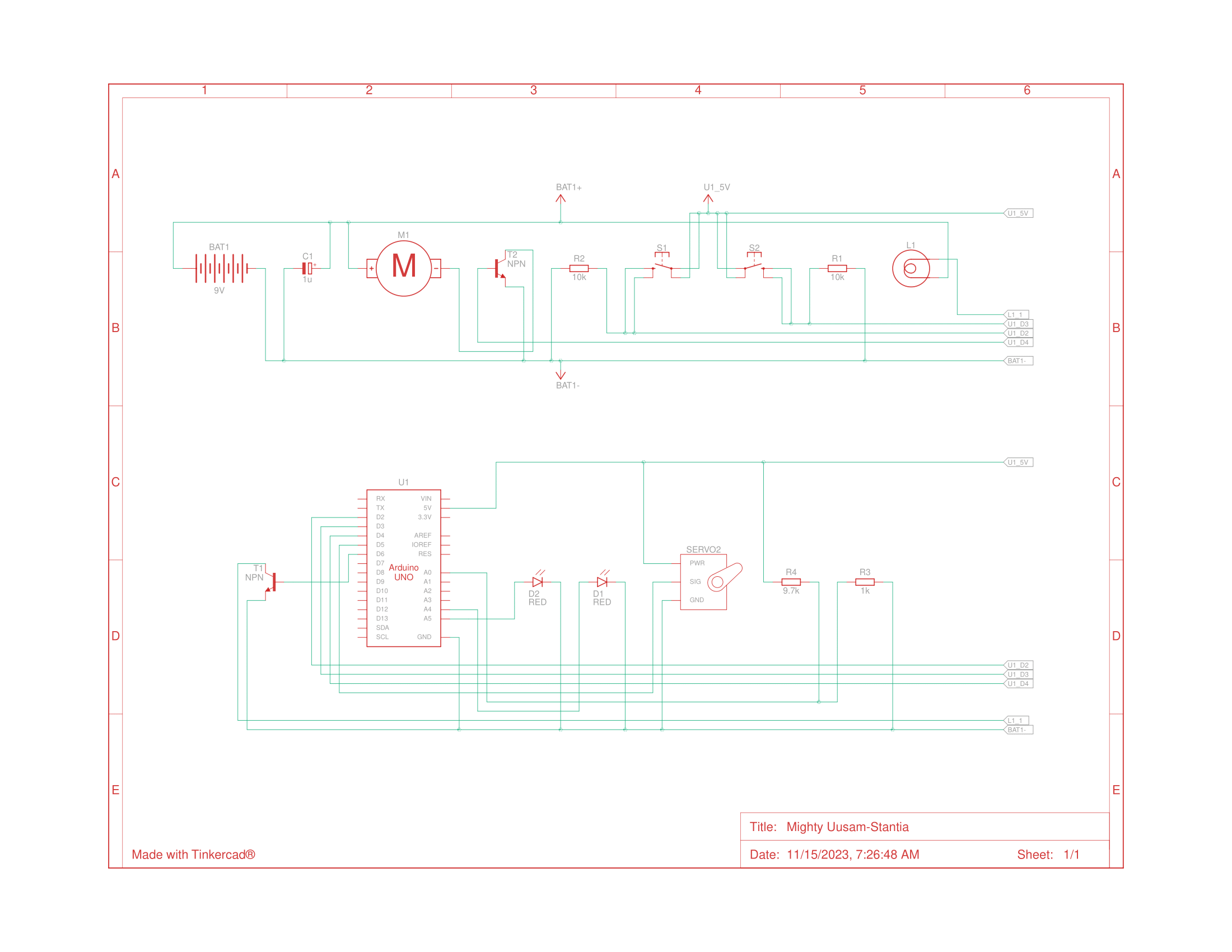
To ensure the reproducibility of experiments and gather additional data, automating the air recycling process within the enclosed space is deemed necessary. To accomplish this task, an Arduino Nano is employed. It systematically collects temperature data and, based on a predefined timer, triggers a valve to evacuate the existing air, enabling the repetition of the experiment with fresh air. It also maintains a constant temperature inside.

There’s the schematic of the mechanism used and a visual representation.



**The software algorithm** is structured around a cyclic process within the Arduino code. In its current configuration, the servo motor opens by 90 degrees every 30 seconds, initiating the operation of the DC motor and LED for a duration of 5 seconds. Subsequently, the servo closes, deactivating both the motor and LED. The system then measures the temperature using the sensor. If the temperature falls below 22 degrees Celsius, the LED remains on until the temperature rises to 23 degrees Celsius. This entire cycle repeats every 30 seconds.

***(Note these values are not the ones used in the experiment, see diagrams below to see the actual data)***

To tailor this algorithm to specific needs, adjustments can be made to parameters such as the angle of servo motor rotation, the duration of motor and Heater activation, the temperature thresholds, and the time intervals for the overall cycle. These modifications offer flexibility in adapting the system to varying experimental requirements and environmental conditions.

See code below.

#define fanPin 4

#define Servo\_PWM 5

#define heaterPin 6

float temp;

int currentTemp = 0;

int targetTemp = 25;

String currentAction = "Before Init";

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <Servo.h>

#include <math.h>

Servo MG995\_Servo;

const int ExhaustTimeMillis = 3000;

const unsigned long CycleTimeMillis = 5000;

unsigned long startTime = 0;

#define SCREEN\_ADDRESS 0x3C

#define OLED\_RESET -1

Adafruit\_SSD1306 display(128, 32, &Wire, -1);

const int temperaturePin = A0;

#define buttonIncrease 2

#define buttonDecrease 3

#define fanPin 4

#define Servo\_PWM 5

#define heaterPin 6

// Define variables

float temp;

int currentTemp = 0;

int targetTemp = 25; // Initial target temperature

String currentAction = "Doing nohing man";

void setup() {

if(!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {

Serial.println(F("SSD1306 allocation failed"));

for(;;); // Don't proceed, loop forever

}

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.clearDisplay();

currentAction = "Initializing";

updateDisplay();

// Initialize buttons

pinMode(buttonIncrease, INPUT\_PULLUP);

pinMode(buttonDecrease, INPUT\_PULLUP);

// Initialize servo, fan, and heater pins

MG995\_Servo.attach(Servo\_PWM);

MG995\_Servo.write(90);

delay(2000);

MG995\_Servo.detach();

pinMode(fanPin, OUTPUT);

pinMode(heaterPin, OUTPUT);

currentAction = "Starting loop";

updateDisplay();

delay(100);

}

void loop() {

Exhaust();

currentAction = "Adjusting temperature";

updateDisplay();

startTime = millis();

while(millis() - startTime < CycleTimeMillis){

// Check and update target temperature

if (digitalRead(buttonIncrease)) {

targetTemp++;

updateDisplay();

}

if (digitalRead(buttonDecrease)) {

targetTemp--;

updateDisplay();

}

if(currentTemp < targetTemp){

analogWrite(heaterPin, 255);

}else{

analogWrite(heaterPin, 0);

}

if(currentTemp > (targetTemp + 2)){

analogWrite(fanPin, 255);

}else{

analogWrite(fanPin, 0);

}

temp = analogRead(temperaturePin);

float v = (4.44 \* temp) / 1023;

float R = 9.7 \* (1 / ((4.44 \* temp / v) - 1));

currentTemp = ConvertOhmToTemp(R);

}

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

}

void Exhaust(){

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Opening Valve";

updateDisplay();

MG995\_Servo.write(0);

delay(3000);

MG995\_Servo.detach();

currentAction = "Act+ Fan and Heater";

updateDisplay();

analogWrite(fanPin, 255);

analogWrite(heaterPin, 255);

delay(ExhaustTimeMillis);

currentAction = "Deact- Fan and Heater";

updateDisplay();

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Closing Valve";

updateDisplay();

MG995\_Servo.write(90);

delay(3000);

MG995\_Servo.detach();

}

//Calibrated for this specific sensor

float ConvertOhmToTemp(float x) {

return 207.866 - 20.3617 \* log(868.361 \* x - 790.267);

}

void updateDisplay() {

// Update OLED display with current and target temperatures

display.clearDisplay();

display.setCursor(0,0);

display.print("Current Temp: ");

display.print(currentTemp);

display.setCursor(0,11);

display.print("Target Temp: ");

display.println(targetTemp);

display.setCursor(0, 22);

display.println(currentAction);

display.display();

}d

void setup() {

if(!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) {

Serial.println(F("SSD1306 allocation failed"));

for(;;); // Don't proceed, loop forever

}

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.clearDisplay();

currentAction = "Initializing";

updateDisplay();

// Initialize buttons

pinMode(buttonIncrease, INPUT\_PULLUP);

pinMode(buttonDecrease, INPUT\_PULLUP);

// Initialize servo, fan, and heater pins

MG995\_Servo.attach(Servo\_PWM);

MG995\_Servo.write(90);

delay(2000);

MG995\_Servo.detach();

pinMode(fanPin, OUTPUT);

pinMode(heaterPin, OUTPUT);

currentAction = "Starting loop";

updateDisplay();

delay(100);

}

void loop() {

Exhaust();

currentAction = "Adjusting temperature";

updateDisplay();

startTime = millis();

while(millis() - startTime < CycleTimeMillis){

// Check and update target temperature

if (digitalRead(buttonIncrease)) {

targetTemp++;

updateDisplay();

}

if (digitalRead(buttonDecrease)) {

targetTemp--;

updateDisplay();

}

if(currentTemp < targetTemp){

analogWrite(heaterPin, 255);

}else{

analogWrite(heaterPin, 0);

}

if(currentTemp > (targetTemp + 2)){

analogWrite(fanPin, 255);

}else{

analogWrite(fanPin, 0);

}

temp = analogRead(temperaturePin);

float v = (4.44 \* temp) / 1023;

float R = 9.7 \* (1 / ((4.44 \* temp / v) - 1));

currentTemp = ConvertOhmToTemp(R);

}

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

}

void Exhaust(){

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Opening Valve";

updateDisplay();

MG995\_Servo.write(0);

delay(3000);

MG995\_Servo.detach();

currentAction = "Act+ Fan and Heater";

updateDisplay();

analogWrite(fanPin, 255);

analogWrite(heaterPin, 255);

delay(ExhaustTimeMillis);

currentAction = "Deact- Fan and Heater";

updateDisplay();

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Closing Valve";

updateDisplay();

MG995\_Servo.write(90);

delay(3000);

MG995\_Servo.detach();

}

//Calibrated for this specific sensor

float ConvertOhmToTemp(float x) {

return 207.866 - 20.3617 \* log(868.361 \* x - 790.267);

}

void updateDisplay() {

// Update OLED display with current and target temperatures

display.clearDisplay();

display.setCursor(0,0);

display.print("Current Temp: ");

display.print(currentTemp);

display.setCursor(0,11);

display.print("Target Temp: ");

display.println(targetTemp);

display.setCursor(0, 22);

display.println(currentAction);

display.display();

}d

targetTemp++;

updateDisplay();

}

if (digitalRead(buttonDecrease)) {

targetTemp--;

updateDisplay();

}

if(currentTemp < targetTemp){

analogWrite(heaterPin, 255);

}else{

analogWrite(heaterPin, 0);

}

if(currentTemp > (targetTemp + 2)){

analogWrite(fanPin, 255);

}else{

analogWrite(fanPin, 0);

}

temp = analogRead(temperaturePin);

float v = (4.44 \* temp) / 1023;

float R = 9.7 \* (1 / ((4.44 \* temp / v) - 1));

currentTemp = ConvertOhmToTemp(R);

}

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

}

void Exhaust(){

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Opening Valve";

updateDisplay();

MG995\_Servo.write(0);

delay(3000);

MG995\_Servo.detach();

currentAction = "Act+ Fan and Heater";

updateDisplay();

analogWrite(fanPin, 255);

analogWrite(heaterPin, 255);

delay(ExhaustTimeMillis);

currentAction = "Deact- Fan and Heater";

updateDisplay();

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Closing Valve";

updateDisplay();

MG995\_Servo.write(90);

delay(3000);

MG995\_Servo.detach();

}

//Calibrated for this specific sensor

float ConvertOhmToTemp(float x) {

return 207.866 - 20.3617 \* log(868.361 \* x - 790.267);

}

void updateDisplay() {

// Update OLED display with current and target temperatures

display.clearDisplay();

display.setCursor(0,0);

display.print("Current Temp: ");

display.print(currentTemp);

display.setCursor(0,11);

display.print("Target Temp: ");

display.println(targetTemp);

display.setCursor(0, 22);

display.println(currentAction);

display.display();

}d

analogWrite(fanPin, 255);

analogWrite(heaterPin, 255);

delay(ExhaustTimeMillis);

currentAction = "Deact- Fan and Heater";

updateDisplay();

analogWrite(fanPin, 0);

analogWrite(heaterPin, 0);

MG995\_Servo.attach(Servo\_PWM);

currentAction = "Closing Valve";

updateDisplay();

MG995\_Servo.write(90);

delay(3000);

MG995\_Servo.detach();

}

//Calibrated for this specific sensor

float ConvertOhmToTemp(float x) {

return 207.866 - 20.3617 \* log(868.361 \* x - 790.267);

}

void updateDisplay() {

// Update OLED display with current and target temperatures

display.clearDisplay();

display.setCursor(0,0);

display.print("Current Temp: ");

display.print(currentTemp);

display.setCursor(0,11);

display.print("Target Temp: ");

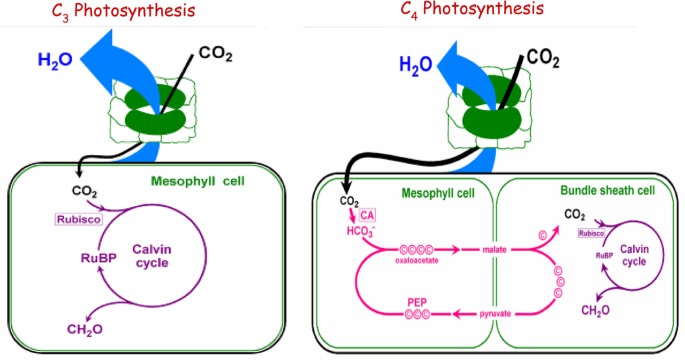
display.println(targetTemp);

display.setCursor(0, 22);

display.println(currentAction);

display.display();

}

**Research subjects:**

*(****A schematic diagram of C3 and C4 photosynthesis****.)*

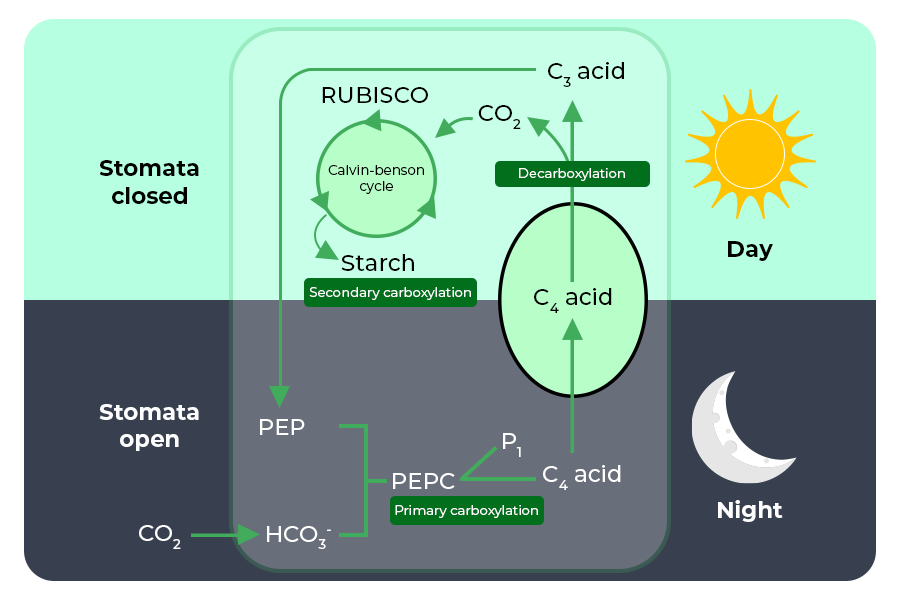
**C3 Photosynthesis:**

1. **Location:** Occurs in the mesophyll cells of most plants.
2. **Key Molecule:** The initial carbon fixation occurs through ribulose-1,5-bisphosphate carboxylase/oxygenase (RuBisCO).
3. **Structural Feature:** C3 plants lack a specific anatomical separation between the light-dependent reactions (calvin cycle) and the dark reactions (RuBisCO).
4. **Working:** CO2 is initially fixed to form a 3-carbon compound, phosphoglycerate. During the Calvin cycle, ATP and NADPH generated in the light reactions are used to convert phosphoglycerate into sugars (glucose).
5. **Efficiency:** C3 plants are generally less efficient in hot, dry conditions due to a tendency of RuBisCO to bind to oxygen instead of CO2 (photorespiration).

**C4 Photosynthesis:**

* **Location:** Occurs in the mesophyll and bundle sheath cells in specific plants adapted to hot and arid environments.
* **Key Molecule:** Phosphoenolpyruvate carboxylase (PEPCase) initially fixes CO2 into a 4-carbon compound, oxaloacetate.
* **Structural Feature:** C4 plants have specialized leaf anatomy with Kranz anatomy, involving mesophyll and bundle sheath cells.
* **Working:** In the mesophyll cells, CO2 is initially fixed into a 4-carbon compound, oxaloacetate, and then transported to bundle sheath cells, where it releases CO2 for the Calvin cycle. This spatial separation reduces photorespiration.
* **Efficiency:** C4 plants are more efficient in hot, dry conditions as they minimize water loss and can maintain photosynthesis with lower stomatal openings.

Crassulacean Acid Metabolism (CAM) plants are a unique group of plants with a distinctive photosynthetic cycle adapted to arid and semi-arid environments. Unlike C3 and C4 plants, CAM plants open their stomata at night to minimize water loss through transpiration. During this nocturnal phase, they uptake carbon dioxide, converting it into organic acids and storing it in vacuoles.

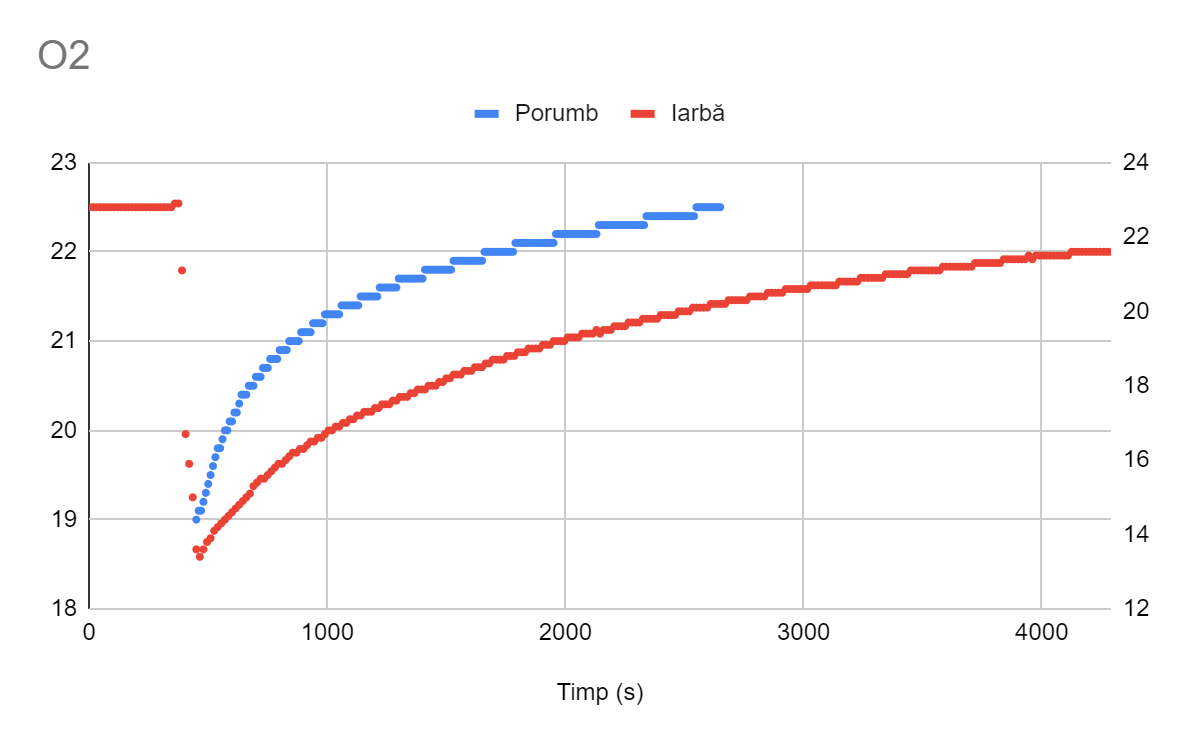
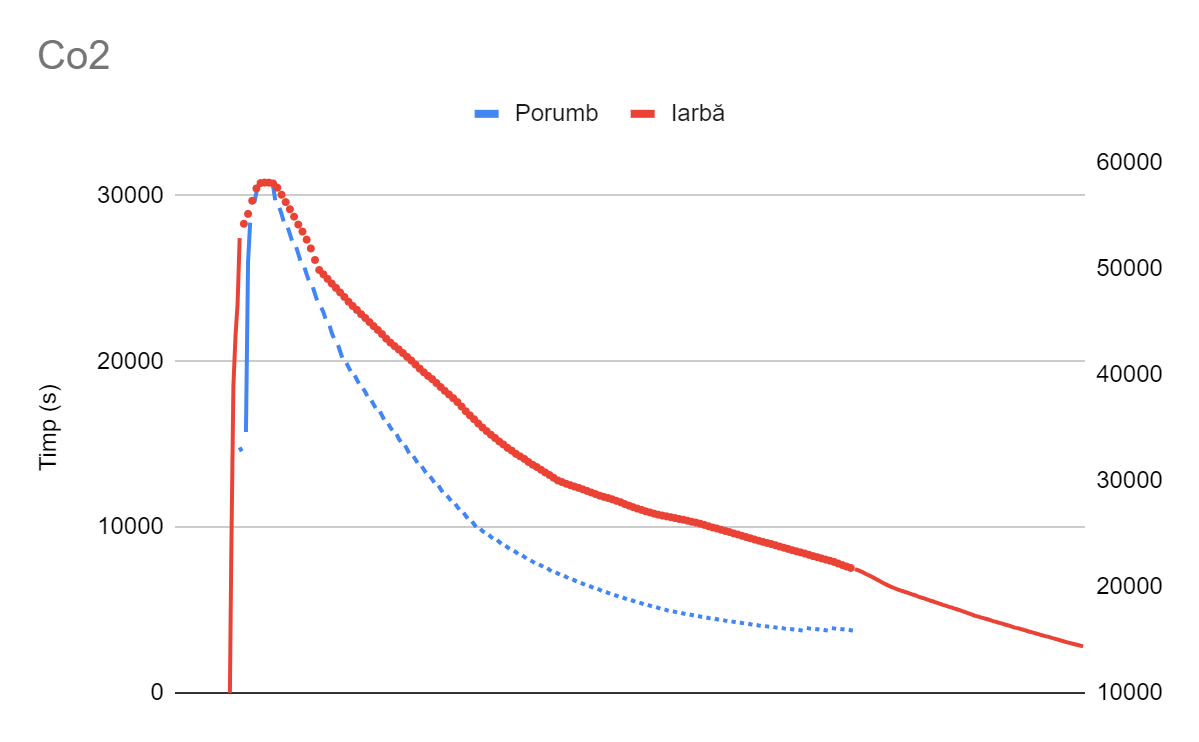


Throughout the day, when stomata are closed to reduce water loss, CAM plants utilize the stored organic acids to conduct photosynthesis. This innovative adaptation allows CAM plants to thrive in water-scarce regions, as they can efficiently capture carbon dioxide during cooler nighttime conditions and perform photosynthesis during the day without excessive water loss. The CAM photosynthetic cycle exemplifies a remarkable strategy for plants to navigate challenging environmental conditions.



Setup and experimentation.

Get data from   
<https://aliph-null.github.io/Data/Project%20Pasco/Study%20Data%20from%20Sparkvue.rar>



**Conclusion**

In conclusion, the data presented in our study strongly supports the notion that C4 plants exhibit greater efficiency compared to C3 plants in various controlled environments. The distinct photosynthetic pathway of C4 plants, characterized by a spatial separation of carbon fixation and the Calvin cycle, enhances their ability to thrive in conditions with elevated temperatures and increased carbon dioxide concentrations. This separation minimizes photorespiration, contributing to improved overall photosynthetic efficiency.

Furthermore, our findings suggest that CAM plants could be considered advantageous for indoor use. Their unique ability to uptake carbon dioxide at night aligns well with indoor environments, potentially enhancing air quality. This aspect makes CAM plants a compelling option for indoor cultivation, providing a dual benefit of efficient photosynthesis and positive air impact.

The validity of our conclusions is reinforced by the insights from two referenced papers. The study by Wang et al. (2012) systematically compares C3 and C4 plants based on metabolic network analysis, providing additional scientific evidence supporting the superior efficiency of C4 plants. Additionally, Dazhong Dave Zhao's work on challenges and approaches to crop improvement through C3-to-C4 engineering further underscores the significance of C4 photosynthesis in enhancing crop performance.

In light of these findings, our research contributes to the growing body of knowledge supporting the superiority of C4 plants in terms of photosynthetic efficiency and highlights the potential benefits of CAM plants for indoor applications.

**Application of findings**

Applying the insights gained from our study through bioengineering and selective breeding presents a promising avenue for creating crops and plants that are not only more efficient but also resilient in arid environments. By leveraging the advantages of C4 photosynthesis, we can engineer or breed crops with enhanced water and resource-use efficiency, making them well-suited for regions facing water scarcity and challenging growing conditions.

Bioengineering efforts could focus on introducing key features of C4 photosynthesis into C3 crops, thereby improving their photosynthetic performance. This may involve modifying enzymes, optimizing cellular structures, and enhancing the overall efficiency of carbon fixation. Selective breeding, on the other hand, can be employed to naturally select plants with inherent traits conducive to C4-like efficiency, promoting the development of crops adapted to arid climates.

The potential impact of these advancements on addressing the world's nutrition problem is substantial. Crops engineered or bred for increased efficiency and resilience could yield higher quantities of nutritious produce, even in regions with limited water resources. This has the potential to bolster food security, mitigate the effects of climate change on agriculture, and provide sustainable solutions for feeding the growing global population.

In essence, the application of our findings through bioengineering and selective breeding holds the key to developing crops that not only survive in arid environments but also contribute significantly to solving the world's nutrition challenges. This approach aligns with the broader goal of creating a more sustainable and resilient agricultural system to meet the demands of an ever-expanding global population.

**Resources, datasheets and references:**

Pasco temperature sensor: <https://www.pasco.com/products/sensors/temperature/ps-3222>

Pasco light sensor: <https://www.pasco.com/products/item/ps-3213>

Pasco CO2 sensor: <https://www.pasco.com/products/sensors/wireless/ps-3208>

Pasco O2 sensor: <https://www.pasco.com/products/sensors/wireless/ps-3217>

Ardiono Nano: <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>

Wang, C., Guo, L., Li, Y. *et al.* Systematic Comparison of C3 and C4 Plants Based on Metabolic Network Analysis. *BMC Syst Biol* **6** (Suppl 2), S9 (2012). <https://doi.org/10.1186/1752-0509-6-S2-S9>

Challenges and Approaches to Crop Improvement Through C3-to-C4 Engineering by Dazhong Dave Zhao, University of Wisconsin–Milwaukee, United States <https://www.frontiersin.org/articles/10.3389/fpls.2021.715391/full>