



**UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA**

Floarea Robotică Inteligentă cu Tracking Solar

Proiectare cu Microprocesoare

Autori: Roman Laura Ioana și Stoleru Alexia Ioana

Grupa: 30235

FACULTATEA DE AUTOMATICĂ
ȘI CALCULATOARE

1 Decembrie 2025

Cuprins

1	Introducere	2
1.1	Cerința proiectului	2
1.2	Problema rezolvată	2
1.3	Soluția propusă	2
1.4	Cerințe atinse (High Level)	3
2	Componente Hardware și Conexiuni	4
2.1	Lista componentelor	4
2.2	Descrierea detaliată a componentelor	4
2.2.1	Arduino UNO	4
2.2.2	PCA9685 - 16-Channel PWM Driver	4
2.2.3	Servomotoare 180° și 360°	4
2.2.4	Senzorul Ultrasonic HC-SR04	5
2.2.5	Fotorezistori	5
2.3	Schema de conexiuni	5
2.4	Considerații Hardware	6
2.4.1	Gestionarea Cerințelor de Putere și a Servomotoarelor	6
2.4.2	Integrarea Senzorilor și Conversia Datelor	6
2.4.3	Flexibilitate și Modularitate	6
3	Software și implementare	6
3.1	Arhitectura Software	6
3.1.1	Librării utilizate	6
3.2	Configurare și Inițializare	7
3.2.1	Definirea constantelor hardware	7
3.2.2	Parametri de calibrare	7
3.2.3	Funcția setup()	7
3.3	Bucloa Principală	8
3.4	Măsurarea Distanței	8
3.5	Citirea Senzorilor de Lumină	9
3.6	Controlul Petalelor	9
3.6.1	Logica de decizie	9
3.6.2	Mișcarea progresivă	9
3.7	Tracking Solar	10
4	Testare	10
4.1	Mod de Funcționare	10
4.1.1	Controlul Petalelor prin Proximitate (Ultrasonic)	11
4.1.2	Orientarea Bazei prin Lumină (Fototropism)	11

1 Introducere

1.1 Cerința proiectului

Pentru acest proiect, am avut ca obiectiv să creăm un sistem robotizat care să imite comportamentul plantelor reale mai exact capacitatea lor de a se orienta către sursele de lumină.

Cerințele principale ale proiectului au fost:

- Implementarea unui mecanism automat de urmărire a surselor de lumină utilizând senzori optici
- Crearea unui sistem de interacțiune cu utilizatorii prin detectarea proximității
- Realizarea unui control coordonat al mai multor actuatori (servomotoare) pentru simularea mișcărilor naturale
- Dezvoltarea unei arhitecturi hardware modulare și extensibile bazate pe platforma Arduino

Proiectul a fost conceput ca o demonstrație practică a principiilor de robotică bio-inspirată, integrând concepte de senzorică, actuare, control și procesare embedded într-un sistem funcțional și interactiv.

1.2 Problema rezolvată

În natură, multe plante prezintă fenomenul de heliotropism: capacitatea de a-și orienta organele vegetative (frunze, flori) către sursele de lumină pentru maximizarea fotosintezei. Această adaptare evolutivă permite plantelor să optimizeze captarea energiei solare, esențială pentru supraviețuire și creștere.

Recrearea acestui comportament într-un sistem robotic ne-a pus mai multe provocări tehnice:

Detectarea direcției luminii: Spre deosebire de sistemele biologice care dispun de fotoreceptori complecși, un sistem artificial trebuie să folosească senzori simpli pentru a determina direcția și intensitatea luminii din mediul înconjurător.

Procesarea informației senzoriale: Datele provenite de la senzori trebuie filtrate, comparate și interpretate pentru a lua decizii corecte de orientare, evitând oscilațiile sau mișcărilor eronate.

Controlul mecanic: Sistemul trebuie să coordoneze mișcarea mai multor actuatori pentru a roti structura către sursă și pentru a simula comportamente naturale precum deschiderea și închiderea petalelor.

Interacțiunea cu mediul: Pe lângă tracking-ul solar, sistemul trebuie să răspundă și la prezența utilizatorilor, creând o experiență interactivă și oferind un comportament mai complex, asemănător organismelor vii.

Soluționarea acestor provocări ne-a permis nu doar crearea unui sistem demonstrativ atractiv, ci și explorarea conceptelor fundamentale din domenii precum robotica bio-inspirată, sistemele embedded și automatizarea proceselor bazate pe senzori.

1.3 Soluția propusă

Pentru rezolvarea problemelor prezentate, am dezvoltat un sistem robotic integrat bazat pe microcontrollerul Arduino UNO, care combină multiple componente hardware și software pentru a realiza comportamentul dorit.

Arhitectura hardware a sistemului este construită în jurul mai multor componente principale care vor fi descrise detaliat în secțiunea următoare: platforma de control Arduino UNO,

driver-ul PWM PCA9685 pentru controlul servomotorilor, doi fotorezistori pentru detectarea luminii, senzor ultrasonic pentru măsurarea distanței și cinci servomotoare pentru actuale.

Arhitectura software implementează doi algoritmi principali care funcționează în paralel:

Algoritmul de tracking solar compară continuu valorile citite de la cei doi fotorezistori și ajustează orientarea structurii către direcția cu lumină mai intensă când diferența depășește un prag predefinit.

Algoritmul de control al petalelor monitorizează distanța măsurată de senzorul ultrasonic și comandă închiderea sau deschiderea lină a celor 4 petale în funcție de proximitatea utilizatorilor.

Ambele subsisteme funcționează independent și simultan, demonstrând capacitatea sistemului de multi-tasking. Comunicația între componente este realizată prin protocol I2C și ADC pe 10 biți, cu interfață Serial pentru debugging.

Implementarea detaliată a algoritmilor și logica de control vor fi prezentate în secțiunea dedicată software-ului.

1.4 Cerințe atinse (High Level)

La finalizarea proiectului, toate cerințele funcționale și tehnice stabilite inițial au fost îndeplinite cu succes:

Funcționalități implementate:

1. Tracking solar automatizat

Sistemul detectează și urmărește autonom sursele de lumină din mediul înconjurător prin rotație continuă. Floarea se orientează progresiv către direcția cu cea mai mare intensitate luminoasă, oprindu-se când ajunge la poziția optimă. Algoritmul implementat previne oscilațiile nedorite și asigură stabilitatea poziției finale.

2. Interacțiune bazată pe proximitate

La detectarea unei persoane în raza de 150 cm, cele patru petale se închid sincronizat într-o mișcare lină de aproximativ 3-4 secunde. La îndepărtarea utilizatorului, petalele se redeschid automat.

3. Mișcare fluidă și naturală

Prin implementarea de loop-uri incrementale cu delay-uri controlate, toate mișcările servomotorilor sunt lente și progresive, eliminând tranziții bruste care ar reduce realismul comportamentului bio-inspirat.

Obiective tehnice atinse:

- Integrare reușită a comunicației I2C pentru control PWM multi-canal
- Implementare corectă a algoritmilor de procesare senzorială
- Sincronizare precisă a actuatorilor multipli
- Calibrare eficientă a pragurilor și parametrilor de control
- Gestionarea corectă a alimentării pentru sistem cu consum ridicat

Prin realizarea tuturor acestor cerințe, proiectul demonstrează nu doar funcționalitatea individuală a componentelor, ci și capacitatea noastră de a integra multiple subsisteme într-un sistem coerent, robust și interactiv.

2 Componente Hardware și Conexiuni

2.1 Lista componentelor

Proiectul implementează funcționalitățile dorite utilizând o selecție specifică de componente hardware esențiale:

- Placă de Dezvoltare: O placă Arduino Uno R3, care servește ca unitate de control centrală.
- Acționare Servo: Un driver cu 16 canale (probabil PCA9685 sau similar) pentru controlul precis al servomotoarelor.
- Servomotoare: Un total de cinci servomotoare, împărțite astfel:
 - 4 servomotoare de 180° (pentru poziționare cu unghi limitat).
 - 1 servomotor de 360° (pentru rotație continuă).
- Senzor de Distanță: Un senzor ultrasonic HC-SR04 pentru măsurarea distanței și detecția obiectelor.
- Sensori de Lumină: Doi fotorezistori (LDR-uri), folosiți pentru detectarea intensității luminoase.

2.2 Descrierea detaliată a componentelor

2.2.1 Arduino UNO

Placa **Arduino Uno R3** reprezintă microcontrolerul central al proiectului, servind ca un „creier” care procesează datele de la senzori și trimite comenzi către actuatori. Echipată cu un microcontroler ATmega328P, aceasta oferă o interfață versatilă cu **14 pini digitali I/O** și **6 pini analogici**, esențiali pentru conectarea senzorilor și a altor periferice. Simplitatea programării sale prin IDE-ul Arduino și vasta comunitate de suport fac din Uno R3 o alegere ideală pentru proiectele prototip, asigurând o execuție stabilă a logicii de control, de la citirea fotorezistorilor la calculul distanței și coordonarea mișcării servomotoarelor.

2.2.2 PCA9685 - 16-Channel PWM Driver

Deoarece placa Arduino Uno are un număr limitat de ieșiri PWM și o capacitate redusă de a furniza curent, se utilizează un **Driver PWM cu 16 canale** (de regulă, un modul bazat pe cipul PCA9685). Această componentă este crucială pentru a extinde capacitatea de control, gestionând independent cele cinci servomotoare necesare. Comunicarea cu Arduino se realizează eficient prin protocolul **I2C (Inter-Integrated Circuit)**, necesitând doar doi pini de pe Arduino (SDA și SCL). Driverul permite setarea precisă a frecvenței PWM și a ciclului de lucru pentru fiecare dintre cele 16 canale, asigurând poziționarea exactă și controlul vitezei servomotoarelor de 180° și 360° fără a supraîncărca microcontrolerul principal.

2.2.3 Servomotoare 180° și 360°

Proiectul se bazează pe un ansamblu de **cinci servomotoare** pentru a executa mișcarea specifică a florii robotice. **Cele patru servomotoare de 180°** sunt dedicate exclusiv controlului mecanic al petalelor. Acestea permit mișcări de poziționare precise și limitate, fiind acționate prin semnale PWM pentru a determina unghiul exact de rotație. Prin variația controlată a unghiului, aceste servomotoare reușesc să creeze **iluzia de deschidere și închidere a petalelor**, imitând astfel comportamentul unei flori naturale.

În contrast, **servomotorul de 360°** (cu rotație continuă) este poziționat la baza mecanismului și are rolul de a orienta întreaga floare robotică. Acesta nu este folosit pentru a seta o poziție

unghiulară fixă, ci pentru a controla **viteza și direcția de rotație continuă**. Acest servomotor este esențial pentru funcționalitatea de urmărire a luminii (phototropism), permițând bazei să se rotească constant și lin, în funcție de citirile senzorilor de lumină (fotorezistorii), asigurând astfel alinierea optimă a florii față de sursa luminoasă dominantă.

2.2.4 Senzorul Ultrasonic HC-SR04

Senzorul **Ultrasonic HC-SR04** este integrat pentru a detecta distanța față de obiecte sau utilizatori în proximitatea florii robotice. Funcționarea sa se bazează pe principiul **sonarului**: microcontrolerul trimite un semnal de 5V pe pinul **Trigger**, care inițiază emiterea unei serii de unde sonore de înaltă frecvență. Aceste unde parcurg distanța până la cel mai apropiat obiect, se reflectă și sunt captate de pinul **Echo** al senzorului.

Arduino măsoară **durata de timp** (Δt) scursă între emisie și recepție. Pe baza acestei durate și a vitezei sunetului, se calculează distanța. Această măsurătoare de proximitate este utilizată ca **parametru de decizie** pentru mecanismul florii. Mai precis, senzorul determină momentul în care petalele trebuie să fie acționate (deschise sau închise), asociind o anumită valoare de distanță cu starea dorită a florii.

2.2.5 Fotorezistori

Cei doi **fotorezistori** (Light-Dependent Resistors sau LDR) sunt senzorii de lumină ai proiectului, utilizați pentru a detecta intensitatea luminoasă și, crucial, **direcția** din care provine cea mai puternică sursă de lumină. Un fotorezistor este un rezistor a cărui rezistență electrică scade pe măsură ce intensitatea luminii incidente crește.

Fiecare LDR este integrat într-un **circuit divizor de tensiune** și conectat la câte un pin analogic al plăcii Arduino. Prin citirea tensiunii de la punctul de mijloc al divizorului, Arduino obține o valoare digitală care este invers proporțională cu luminozitatea. Poziționând cele două LDR-uri la o distanță mică unul de celălalt (de exemplu, sub un vizor direcțional), se poate compara intensitatea luminii primite de fiecare. Diferența dintre cele două citiri indică direcția exactă în care trebuie rotit mecanismul (folosind servomotoarele) pentru a se alinia perfect cu sursa de lumină.

2.3 Schema de conexiuni

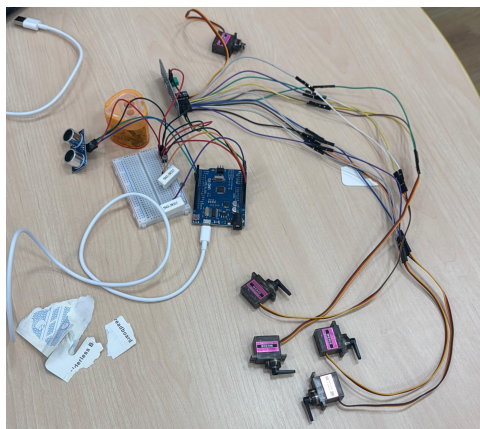


Figura 1: Schema

2.4 Considerații Hardware

Secțiunea consideră aspectele critice care au influențat selecția și arhitectura sistemului hardware, concentrându-se pe gestionarea energiei și extinderea capacităților de control ale plăcii Arduino.

2.4.1 Gestionarea Cerințelor de Putere și a Servomotoarelor

O provocare majoră a fost asigurarea energiei necesare pentru acționarea simultană și stabilă a celor cinci servomotoare. Placa **Arduino Uno R3** nu poate furniza curentul necesar prin porturile sale, riscând instabilitatea sau resetarea. Prin urmare, a fost necesară implementarea unei **sursă de alimentare externă** dedicată exclusiv servomotoarelor. Pentru a rezolva problema numărului limitat de pini PWM disponibili pe Arduino, s-a optat pentru utilizarea unui **driver cu 16 canale (e.g., PCA9685)**. Acest driver comunică cu Arduino prin interfața **I2C**, utilizând doar doi pini (SDA și SCL), dar oferind 16 ieșiri PWM dedicate, asigurând astfel controlul precis al poziției petalelor (180°) și al mișcării bazei (360°).

2.4.2 Integrarea Senzorilor și Conversia Datelor

Integrarea senzorilor a impus o schemă mixtă de conexiuni. Senzorul ultrasonic **HC-SR04** a fost conectat la pinii digitali pentru funcționarea de tip I/O (Trigger/Echo), în timp ce cei **doi fotorezistori** au necesitat pinii analogici (A0, A1). Această configurare analogică a necesitat includerea fiecărui fotorezistor într-un **circuit divizor de tensiune**.

2.4.3 Flexibilitate și Modularitate

Alegerea componentelor a fost ghidată și de principiul **modularității**. Utilizarea driverului I2C nu doar că a rezolvat problema lipsei de pini, dar a creat și un sistem scalabil. În cazul unei extinderi viitoare a proiectului (de exemplu, adăugarea de lumini LED sau mai multe servomotoare), driverul permite conectarea a încă 11 actuatoare fără a schimba placa de bază. De asemenea, utilizarea plăcii **Arduino Uno R3** a oferit o platformă robustă și accesibilă pentru dezvoltarea rapidă a algoritmilor de control.

3 Software și implementare

3.1 Arhitectura Software

Codul proiectului este structurat modular, organizat în funcții independente care gestionează subsisteme specifice ale florii robotice. Programul este bazat pe un loop principal care execută ciclic operații de citire senzori, procesare date și control.

3.1.1 Librării utilizate

Pentru implementare am utilizat două librării esențiale:

Wire.h - Librăria standard Arduino pentru comunicație I2C, necesară pentru comunicarea cu driver-ul PCA9685. Protocolul I2C folosește doar două fire (SDA și SCL), reducând numărul de conexiuni necesare.

Adafruit_PWMServoDriver.h - Librărie pentru controlul driver-ului PCA9685, oferind funcții simplificate pentru configurarea frecvenței PWM și trimiterea comenzilor către cele 16 canale.

3.2 Configurare și Inițializare

3.2.1 Definirea constantelor hardware

La începutul codului am definit constantele care mapează componentele hardware:

```
const int TRIG_PIN = 6;           // Pin trigger ultrasonic
const int ECHO_PIN = 7;           // Pin echo ultrasonic
const int LDR_LEFT = A0;          // Fotorezistor stânga
const int LDR_RIGHT = A1;         // Fotorezistor dreapta
const int ROTATION_SERVO_CHANNEL = 4; // Canal PWM servo rotație
```

Senzorul ultrasonic folosește pinii digitali 6 și 7, în timp ce fotorezistorii sunt conectați la pinii analogici A0 și A1 pentru citirea valorilor de tensiune.

3.2.2 Parametri de calibrare

Am definit constantele care controlează comportamentul sistemului, ajustate experimental:

```
const int DISTANCE_THRESHOLD = 150; // Prag distanță (cm)
const int LIGHT_THRESHOLD = 20;     // Diferență minimă lumină
const int ROTATION_DURATION = 800;  // Durată rotație (ms)
const int SERVO_SPEED_OFFSET = 25;  // Offset viteză rotație
int SERVO_STOP = 315;               // Valoare STOP calibrată
```

DISTANCE_THRESHOLD = 150 cm oferă o zonă de detecție generoasă pentru interacțiune.

LIGHT_THRESHOLD = 20 previne rotații excesive cauzate de zgomot senzorial. Valori mai mici ar cauza instabilitate.

ROTATION_DURATION = 800 ms oferă cel mai bun compromis între viteză și precizie de ajustare.

SERVO_STOP = 315 a fost determinat prin calibrare experimentală. Fiecare servo continuu are o valoare unică de stop datorită toleranțelor de fabricație.

Pentru servomotoarele petalelor am definit pozițiile extreme:

```
const int PETALS_CLOSED = 125; // Poziție închis
const int PETALS_OPEN = 485;   // Poziție deschis
```

3.2.3 Funcția setup()

Funcția `setup()` se execută o singură dată la pornire și configurează componentele:

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

    pinMode(TRIG_PIN, OUTPUT);
```



```

pinMode(ECHO_PIN, INPUT);
pinMode(LDR_LEFT, INPUT);
pinMode(LDR_RIGHT, INPUT);

pwm.begin();
pwm.setPWMFreq(SERVO_FREQ); // 50Hz pentru servomotoare

pwm.setPWM(ROTATION_SERVO_CHANNEL, 0, SERVO_STOP);
setPetalPosition(PETALS_CLOSED);

delay(2000);
}

```

Setarea inițială a servoul de rotație la SERVO_STOP este critică pentru a preveni rotația necontrolată la pornire.

3.3 Bucla Principală

Funcția `loop()` reprezintă nucleul sistemului și se execută continuu:

```

void loop() {
    currentDistance = measureDistance();
    readLightSensors();
    controlPetals();
    trackLight();
    printDebugInfo();
    delay(100);
}

```

Ordinea operațiilor este importantă: citim senzori, apoi procesăm datele și luăm decizii de control. Delay-ul de 100ms previne execuția prea rapidă și reduce zgomotul senzorial.

3.4 Măsurarea Distanței

Funcția `measureDistance()` implementează protocolul HC-SR04:

```

float measureDistance() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);

    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    float duration_us = pulseIn(ECHO_PIN, HIGH, 30000);

    if (duration_us == 0)
        return 999;
}

```

```

    return (duration_us * 0.0343) / 2;
}

```

Protocolul: se trimite un puls de 10s pe TRIG, senzorul emite ultrasunet, iar ECHO rămâne HIGH până când ecoul revine. Distanța se calculează cu formula $d = \frac{t \times 0.0343}{2}$, unde 0.0343 este viteza sunetului în cm/s, împărțit la 2 deoarece sunetul face dus-întors.

3.5 Citirea Senzorilor de Lumină

```

void readLightSensors() {
    lightLeft = analogRead(LDR_LEFT);
    lightRight = analogRead(LDR_RIGHT);
}

```

Fiecare LDR este conectat într-un divizor de tensiune cu rezistență de 10k. ADC-ul pe 10 biți transformă tensiunea (0-5V) în valori digitale (0-1023). Valori mari indică lumină intensă.

3.6 Controlul Petalelor

3.6.1 Logica de decizie

```

void controlPetals() {
    if (currentDistance < DISTANCE_THRESHOLD && petalsOpen) {
        closePetals();
        petalsOpen = false;
    }
    else if (currentDistance >= DISTANCE_THRESHOLD && !petalsOpen) {
        openPetals();
        petalsOpen = true;
    }
}

```

Flag-ul `petalsOpen` previne deschiderea/închiderea repetată când distanța oscilează în jurul pragului.

3.6.2 Mișcarea progresivă

```

void openPetals() {
    for (int pos = currentPetalPosition; pos <= PETALS_OPEN; pos += 10) {
        setPetalPosition(pos);
        currentPetalPosition = pos;
        delay(50);
    }
}

```

Incremente de 10 unități la fiecare 50ms creează o mișcare fluidă. Diferența de 360 unități (485-125) în 36 pași = 1.8 secunde pentru deschidere completă. Variabila `currentPetalPosition` permite continuarea mișcării de unde a rămas dacă este întreruptă.

Funcția `setPetalPosition()` trimite aceeași comandă la toate cele 4 servomotoare (canale 0-3) pentru sincronizare perfectă.

3.7 Tracking Solar

Funcția `trackLight()` implementează logica de orientare către lumină:

```
void trackLight() {
    static unsigned long lastRotationTime = 0;
    static bool isRotating = false;

    int lightDifference = lightRight - lightLeft;

    if (isRotating) {
        if (millis() - lastRotationTime >= ROTATION_DURATION) {
            pwm.setPWM(ROTATION_SERVO_CHANNEL, 0, SERVO_STOP);
            isRotating = false;
        }
        return;
    }

    if (abs(lightDifference) > LIGHT_THRESHOLD) {
        int speed;

        if (lightDifference > 0) {
            speed = SERVO_STOP + SERVO_SPEED_OFFSET; // Dreapta
        } else {
            speed = SERVO_STOP - SERVO_SPEED_OFFSET; // Stânga
        }

        pwm.setPWM(ROTATION_SERVO_CHANNEL, 0, speed);
        lastRotationTime = millis();
        isRotating = true;
    }
}
```

Variabile statice: Cuvântul cheie `static` asigură că variabilele își păstrează valoarea între apeluri, implementând o mașină de stări pentru gestionarea rotației.

Logica de decizie: Diferența $\Delta L = L_{dreapta} - L_{stânga}$ determină direcția:

- $\Delta L > 20$: rotație dreapta ($\text{speed} = 315 + 25 = 340$)
- $\Delta L < -20$: rotație stânga ($\text{speed} = 315 - 25 = 290$)
- $|\Delta L| \leq 20$: oprire (balanced)

Mecanism de protecție: Când `isRotating` este true, funcția verifică doar dacă au trecut 800ms și oprește rotația. Acest mecanism previne comenzi conflictuale și asigură rotații incrementale controlate.

4 Testare

4.1 Mod de Funcționare

Floarea robotică prezintă două moduri de operare distincte, care pot fi testate și utilizate cu ușurință: controlul petalelor prin proximitate și orientarea automată bazată pe lumină.

4.1.1 Controlul Petalelor prin Proximitate (Ultrasonic)

Această funcționalitate emulează o interacțiune directă cu utilizatorul și depinde de senzorul **HC-SR04**. Atunci când un obiect (de exemplu, mâna utilizatorului) este **apropiat** de senzor, depășind un prag de distanță predefinit, microcontrolerul trimite comenzi către cele patru servomotoare de 180°. Această acțiune determină **deschiderea petalelor**, simulând un gest de răspuns sau „înflorire”. Invers, atunci când obiectul este **îndepărtat** și distanța măsurată crește dincolo de pragul stabilit, servomotoarele inversează mișcarea, provocând **închiderea petalelor** la loc.

4.1.2 Orientarea Bazei prin Lumină (Fototropism)

A doua funcționalitate utilizează cei doi **fotorezistori (LDR)** pentru a imita comportamentul natural de **fototropism**. Principiul de testare implică iluminarea selectivă: prin concentrarea unei surse de lumină (precum lanterna unui telefon) pe **unul dintre fotorezistori**, se creează o diferență semnificativă între citirile celor doi senzori. Algoritmul intern detectează dezechilibrul și comandă servomotorul de 360° din baza florii să se rotească.