

Proiect 2 – Electronica aplicata Panou solar cu urmarire

Studenti:

Serbanescu Daniela-Cristina
Budur Maria

Grupa:

434D

Profesor coordonator:

Neacsu Ana
Guzu Alexandru
Daescu Andrei
Enachescu Dorin

1.	Cuprins	
2.	Introducere.....	3
3.	Resurse hardware	4
4.	Resurse software.....	7
5.	Implementare hardware	11
6.	Implementare software	13
7.	Concluzii	16
8.	Bibliografie.....	16

2. Introducere

În ultimii ani, energia regenerabilă a devenit un domeniu de interes major, atât din perspectiva protejării mediului înconjurător, cât și din cea a dezvoltării durabile. Dintre sursele regenerabile, energia solară este una dintre cele mai accesibile și mai curate forme de producere a energiei electrice. Panourile solare sunt din ce în ce mai răspândite în mediul urban și rural, însă eficiența lor depinde semnificativ de unghiul de incidență al razelor solare. Dacă panoul este orientat perpendicular pe razele solare, cantitatea de energie produsă crește considerabil. Din acest motiv, s-au dezvoltat sisteme inteligente de urmărire a Soarelui, care permit panourilor să-și ajusteze automat poziția în funcție de traiectoria acestuia pe cer.

Proiectul de față urmărește construirea unui astfel de sistem la scară mică, cu scop demonstrativ și educativ. În locul unui panou solar de mari dimensiuni, se utilizează o **celulă fotovoltaică de dimensiuni reduse**, montată pe un suport mobil. Aceasta este **orientată automat în funcție de direcția luminii**, cu ajutorul unui **servomotor**, controlat de o **placă de dezvoltare Arduino**. În acest caz, pentru testarea funcționării sistemului, s-a folosit o **lanternă ca sursă de lumină artificială**, ceea ce permite simularea dinamică a deplasării Soarelui într-un mod accesibil și controlat.

Sistemul este bazat pe doi **senzori de lumină (foto-rezistori sau LDR-uri)**, plasați de o parte și de alta a celulei fotovoltaice. Aceștia detectează intensitatea luminoasă din două direcții apropiate. Atunci când unul dintre senzori primește mai multă lumină decât celălalt, Arduino compară valorile și comandă **servomotorului** să rotească panoul în direcția cu iluminare mai intensă. Astfel, celula fotovoltaică se aliniază automat cu sursa de lumină, imitând comportamentul unui **tracker solar automatizat**.

Servomotorul este o alegere potrivită pentru acest tip de aplicație, întrucât permite **control precis al unghiului de rotație**, fără a necesita senzori suplimentari pentru feedback. Acest lucru simplifică atât partea hardware, cât și logica software. În plus, servomotorul este alimentat și controlat ușor de Arduino prin semnale PWM (modulație în lățime de impuls), ceea ce îl face ideal pentru proiecte de mică amploare.

Proiectul include și un **afișaj LCD**, care oferă utilizatorului informații în timp real despre **poziția curentă a panoului (în grade)** și **valorile de lumină detectate** de fiecare senzor. Această componentă adaugă un plus de interactivitate și ajută la înțelegerea funcționării sistemului în diferite condiții de iluminare.

Etapele de realizare ale acestui proiect includ:

- proiectarea **schemei electrice** și asamblarea componentelor pe breadboard;
- dezvoltarea unui **algoritm de urmărire a luminii**, inițial sub formă de **pseudocod**;
- implementarea codului de control în **limbaj Arduino (bazat pe C++)**;
- testarea sistemului prin deplasarea sursei de lumină (lanterna) și observarea răspunsului automat al servomotorului.

Prin realizarea acestui sistem, se urmărește nu doar crearea unei aplicații funcționale, ci și aprofundarea unor concepte fundamentale de **automatizare, senzori, actuatori și programare embedded**. Utilizarea plăcii Arduino permite o învățare intuitivă și rapidă, datorită comunității extinse și a mediului de dezvoltare accesibil. Acest proiect se încadrează perfect într-un cadru educațional, fiind un exemplu clar de **aplicație practică a teoriei în domeniul energiei regenerabile** și al controlului automat.

În concluzie, acest sistem de urmărire a luminii reprezintă o demonstrație practică a modului în care tehnologia poate fi utilizată pentru a crește eficiența energetică, chiar și în condiții simple. Folosind componente de bază precum Arduino, servomotorul și LDR-urile, proiectul reușește să simuleze un proces real – **urmărirea automată a sursei de lumină** – și oferă o bază solidă pentru viitoare îmbunătățiri, cum ar fi adăugarea unei axe suplimentare de mișcare, integrarea cu senzori de mediu sau utilizarea datelor colectate pentru optimizări suplimentare.

3. Resurse hardware

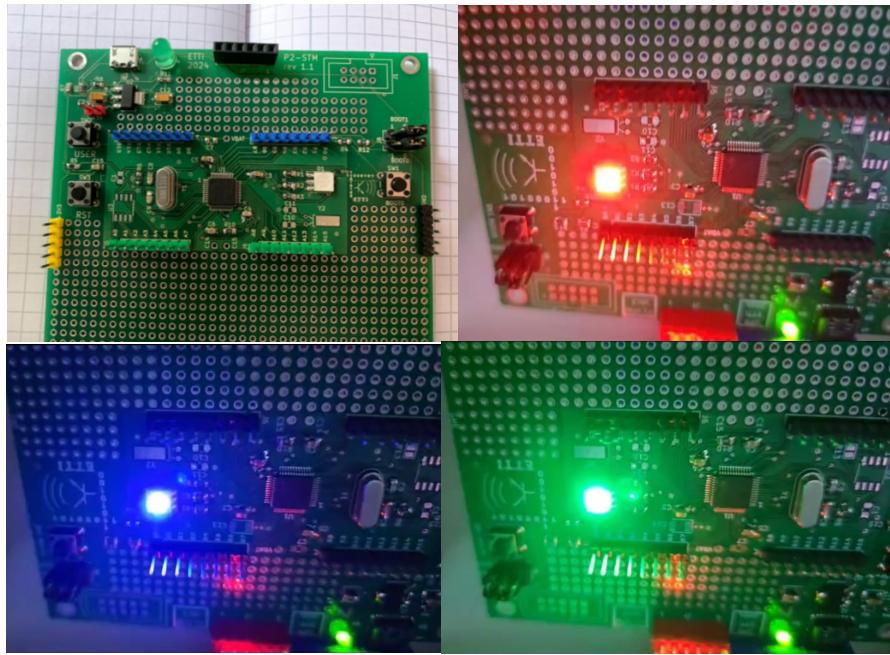


Fig. 3.1 - Placa de dezvoltare STM32

Familia de microcontrolere **STM32F103xx** cu **densitate medie de performanță** integrează nucleul **Arm Cortex-M3 RISC pe 32 de biți**, cu performanțe ridicate, funcționând la o frecvență de **72 MHz**, **memorii rapide integrate** (memorie Flash de până la **128 Kbytes** și **SRAM de până la 20 Kbytes**) și o gamă extinsă de **intrări/ieșiri (I/O)** și **periferice îmbunătățite**, conectate la **două magistrale APB**.

Toate dispozitivele din această familie oferă:

- două **converttoare analog-digitale (ADC)** de **12 biți**,
- trei **temporizatoare generale pe 16 biți**,
- un **temporizator PWM** (pentru generarea semnalelor modulate în lățime),
- și interfețe de comunicație **standard și avansate**, inclusiv:
 - până la **două interfețe I2C și două SPI**,
 - **trei USART**,
 - **un port USB**,
 - și **o interfață CAN**.

Dispozitivele funcționează cu o tensiune de alimentare cuprinsă între **2.0 și 3.6 V** și sunt disponibile în două intervale de temperatură:

- standard: de la **-40°C până la +85°C**,
- extins: de la **-40°C până la +105°C**.

Un set complet de **moduri de economisire a energiei** permite proiectarea de aplicații cu consum redus de putere.

Familia STM32F103xx include **șase tipuri de capsule (package-uri)**, variind de la **36 până la 100 de pini**. În funcție de dispozitivul ales, este disponibil un set diferit de periferice. Descrierea de mai sus oferă o privire generală asupra gamei complete de funcționalități propuse de această familie.

Aceste caracteristici fac ca familia de microcontrolere STM32F103xx cu densitate medie de performanță să fie potrivită pentru o **gama largă de aplicații**, precum:

- **acționări de motoare**,
- **control de aplicații industriale**,
- **echipamente medicale și portabile**,

- periferice pentru PC și gaming,
- platforme GPS,
- automatizări industriale, PLC-uri, invertoare,
- imprimante, scanere,
- sisteme de alarmă, interfoane video,
- și HVAC (sisteme de încălzire, ventilație și aer condiționat).

Placa de dezvoltare **STM32F103C8T6**, cunoscută în mod popular sub numele de **Blue Pill**, este o platformă hardware compactă și versatilă, utilizată pe scară largă în dezvoltarea de aplicații embedded. Aceasta are la bază microcontrolerul **STM32F103C8T6** din familia STM32, produs de compania **STMicroelectronics**, și este apreciată pentru raportul foarte bun între preț, performanță și funcționalitate.

Microcontrolerul STM32F103C8T6 este bazat pe nucleul **ARM Cortex-M3 pe 32 de biți**, care funcționează la o frecvență de **până la 72 MHz**, oferind o capacitate mare de procesare într-un consum redus de energie. Acesta dispune de **64 KB de memorie Flash** pentru stocarea codului și **20 KB de SRAM** pentru date.

Placa oferă o gamă largă de **periferice integrate**, fiind echipată cu:

- **2 convertoare analog-digitale (ADC) de 12 biți**, utile pentru achiziția datelor de la senzori analogici (cum ar fi foto-rezistorii),
- **interfețe de comunicație serială multiple**: 3 × USART, 2 × I2C, 2 × SPI, o interfață USB și o interfață CAN,
- **capabilități PWM** pe mai mulți pini, ideale pentru controlul servomotoarelor,
- și un LED de stare conectat intern la pinul PC13.

Tensiunea de operare a microcontrolerului este cuprinsă între **2.0 V și 3.6 V**, însă placa poate fi alimentată direct la 5V prin conectorul micro-USB, mulțumită regulatorului de tensiune integrat (AMS1117-3.3).

Un avantaj important al plăcii Blue Pill îl reprezintă **conectivitatea GPIO extinsă**, având până la **37 de pini digitali configurați** ca intrări sau ieșiri, în funcție de necesitățile aplicației. Aceștia sunt dispuși în două rânduri laterale, ceea ce permite montarea plăcii pe o **breadboard**, facilitând astfel prototiparea rapidă.

Programarea plăcii se poate realiza în mai multe moduri:

- prin interfața **ST-Link V2** (folosind protocolul SWD – Serial Wire Debug),
- sau prin bootloader-ul integrat, folosind un convertor **USB-UART** (ex. modul CP2102 sau CH340G).

Pentru dezvoltarea aplicațiilor software se pot utiliza instrumente precum **STM32CubeIDE**, **STM32CubeMX** (pentru configurarea pinilor și generarea codului), sau chiar **Arduino IDE**, după instalarea pachetului corespunzător pentru STM32.

Placa STM32F103C8T6 este ideală pentru proiecte de control, automatizare, achiziție de date și comunicație. Datorită dimensiunii reduse, consumului scăzut și performanțelor bune, aceasta este frecvent utilizată în aplicații precum:

- controlul motoarelor și servomotoarelor,
- sisteme de urmărire a luminii sau a Soarelui,
- interfețe cu senzori de temperatură, lumină sau umiditate,
- afișaje LCD/I2C,
- dispozitive portabile sau autonome.

În concluzie, placa **STM32F103C8T6 Blue Pill** este o soluție eficientă și accesibilă pentru dezvoltarea rapidă de sisteme embedded, fiind potrivită atât pentru studenți și pasionați, cât și pentru aplicații prototip sau comerciale de complexitate medie.

Tipuri de conexiuni explicate:

- **5V** – Tensiune de alimentare pentru componente electronice.
- **GND (Ground)** – Referință comună de tensiune (masă electrică).
- **I/O digitali/analogici** – Pini de intrare/ieșire de pe microcontroler:
 - Analogici: citesc semnalul de la foto-rezistori (valori de tensiune).
 - Digitali: controlează motorul, comunică cu LCD-ul, etc.

Pentru realizarea circuitului am ales câteva componente electronice de bază:

1. Breadboard:

Rol: Este o platformă reutilizabilă de testare pentru conectarea componentelor electronice fără a fi nevoie de lipire.

Funcție în proiect: Permite conectarea senzorilor (foto-rezistori), a panoului fotovoltaic, a servomotorului, a afișajului LCD și a plăcii de control (ex. STM32 sau Arduino) într-un circuit complet, ușor de modificat și testat.

2. Celulă fotovoltaică (panou solar):

Rol: Este un dispozitiv care transformă lumina solară în energie electrică.

Funcție în proiect: Nu este folosită pentru generarea de energie în acest caz, ci este montată pe un suport mobil (acționat de un motor), iar sistemul va urmări lumina pentru a simula un panou solar care se orientează singur după Soare (sau lanternă).

3. Foterezistori (LDR - Light Dependent Resistors):

Rol: Sunt senzori care își modifică rezistența electrică în funcție de cantitatea de lumină primită – cu cât lumina este mai puternică, cu atât rezistența este mai mică.

Funcție în proiect: Se folosesc doi foterezistori pentru a compara intensitatea luminii din două direcții. Diferența dintre valorile lor este folosită de microcontroler pentru a decide în ce direcție să se miște panoul pentru a urmări sursa de lumină.

4. LCD (Liquid Crystal Display):
Rol: Este un afișaj electronic care poate fi folosit pentru a arăta text, numere sau informații în timp real.

Funcție în proiect: Este folosit pentru a afișa:

- Unghiul curent de rotație al panoului solar (poziția motorului);
- Valoarea intensității luminii detectate de senzori. Acest lucru ajută utilizatorul să vadă cum răspunde sistemul la modificările luminii din jur.

5. Servomotor:

Rol: Este un motor electric care se mișcă în pași mici și precisi (sau unghiuri controlate, în cazul servomotorului).

Funcție în proiect: Este responsabil pentru rotirea panoului solar spre stânga sau dreapta, în funcție de datele primite de la foto-rezistori. Motorul este comandat de microcontroler (Arduino sau STM32) prin semnale digitale/PWM.

6. Rezistori de 10kΩ:

Rol: Formarea unui divizor de tensiune împreună cu senzorii de lumină (LDR)

Funcție în proiect:

- **Convertesc variațiile de lumină captate de LDR în semnale de tensiune** care pot fi citite de microcontroler (ex: Arduino).
- **Permit compararea intensității luminii pe direcții diferite** pentru a detecta poziția optimă a soarelui.
- **Asigură stabilitate și echilibru** între cele două canale de detecție.

7. Convertor USB-TTL:

Rol: Comunicarea dintre un **microcontroler** (precum STM32) și un **calculator**, în special în etapa de programare, testare sau monitorizare serială. Acesta convertește semnalele **USB (Universal Serial Bus)** de pe calculator în semnale **TTL (Transistor-Transistor Logic)** utilizate de microcontrolere.

Funcție în proiect:

- **Încărcarea codului** în microcontroler (dacă nu există altă interfață de programare disponibilă).
- **Monitorizarea valorilor senzorilor (foterezistorilor)** în timp real, printr-un terminal serial.
- **Diagnosticarea funcționării servomotorului**, pentru a verifica dacă impulsurile PWM sunt generate corect.
- **Configurarea sau calibrarea** sistemului dintr-un software pe PC.

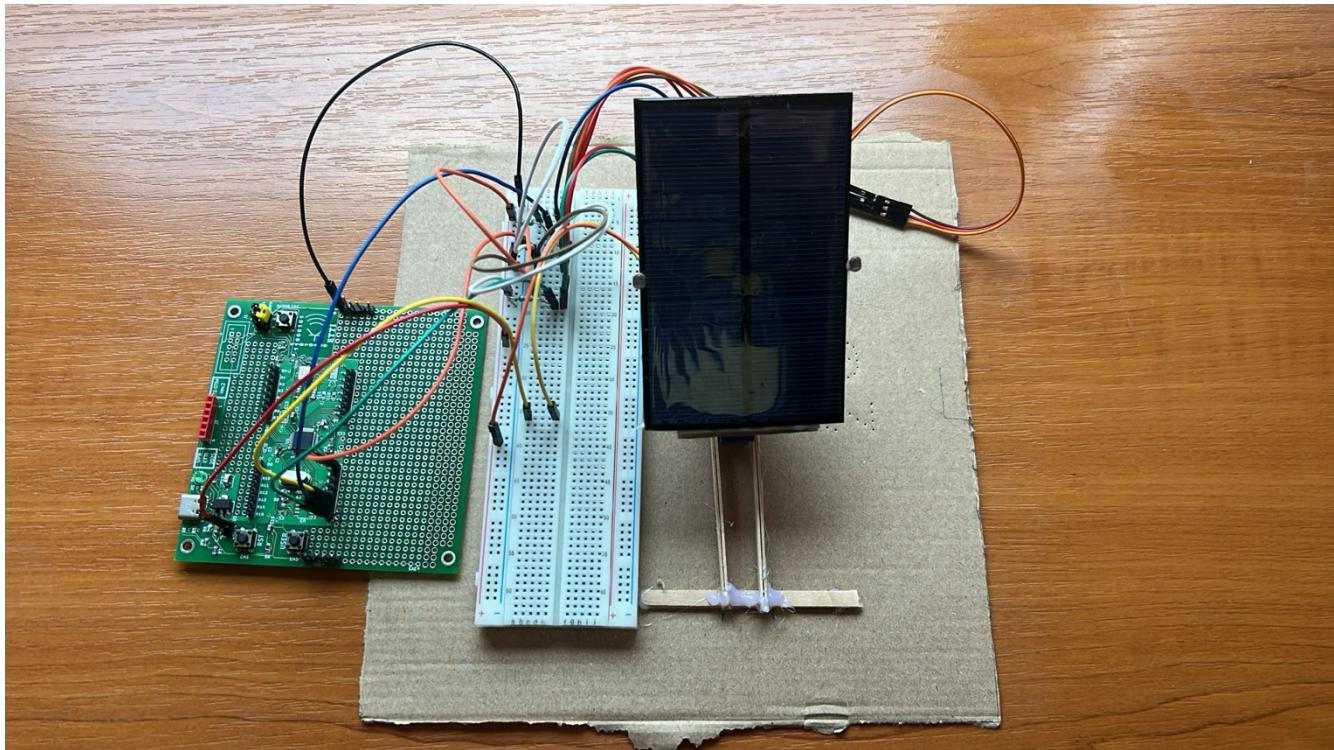


Fig. 3.2 – Circuitul panoului solar

4. Resurse software

În cadrul acestui proiect, au fost utilizate mai multe instrumente software și librării externe pentru a implementa logica de control a sistemului de urmărire a luminii. Acestea au avut un rol esențial în configurarea componentelor electronice, programarea funcționalităților și afișarea valorilor relevante pe ecranul LCD.

1. Mediul de dezvoltare: Arduino IDE

Pentru dezvoltarea codului, a fost utilizat Arduino IDE, un mediu de dezvoltare integrat simplu și intuitiv, dedicat programării microcontrolerelor din familia Arduino. Acest mediu oferă:

- un editor de cod în limbaj C/C++ simplificat,
- funcții predefinite pentru controlul pinilor (ex: analogRead(), digitalWrite()),
- o interfață de monitorizare serială (Serial Monitor) pentru depanare,
- facilități de încărcare directă a codului pe placă prin USB.

Arduino IDE este compatibil cu majoritatea plăcilor de dezvoltare și permite extinderea funcționalităților prin instalarea de biblioteci externe.

2. Librării software utilizate

Pentru controlul componentelor conectate la microcontroler, au fost folosite următoarele librării: Servo.h

- Permite controlul facil al servomotoarelor prin semnale PWM.
- Prin funcția servo.write(unghi);, motorul poate fi poziționat cu precizie la unghiuri între 0° și 180°.
- A fost folosită pentru rotirea celulei fotovoltaice în direcția sursei de lumină.

3. Funcționarea codului și logica software

Codul principal al aplicației a fost scris în limbaj C/C++ folosind funcții și structuri specifice platformei Arduino. Funcționarea generală poate fi descrisă astfel:

- Se citesc valorile de la doi senzori LDR prin intermediul pinilor analogici (analogRead()).
- Se compară valorile celor doi senzori:

- Dacă valoarea din stânga este mai mare decât cea din dreapta, se rotește servomotorul la stânga.
- Dacă valoarea din dreapta este mai mare, se rotește la dreapta.
- Dacă valorile sunt aproximativ egale, servomotorul rămâne în poziție.
- Unghiul actual este actualizat și afișat pe LCD, împreună cu valorile luminozității.
- Se introduce un mic delay (delay()) pentru a evita reacțiile bruște.

4. Depanare și testare

Pentru testare și calibrare, s-a folosit Serial Monitor din Arduino IDE. Acesta a permis afișarea în timp real a valorilor citite de la senzori și verificarea corectitudinii logice a algoritmului de urmărire a luminii.

5. Extensibilitate

Codul este modular și poate fi extins ușor pentru:

- Adăugarea unei a două axe de mișcare (2 servomotoare pentru mișcare X-Y),
- Stocarea valorilor de lumină în EEPROM sau pe card SD,
- Transmiterea datelor către o aplicație mobilă sau pe internet prin module Wi-Fi (ex: ESP8266).

IDE-ul pus la dispoziție de Arduino conține:

- O fereastră principală unde se scrie codul, folosind limbajul C/C++:

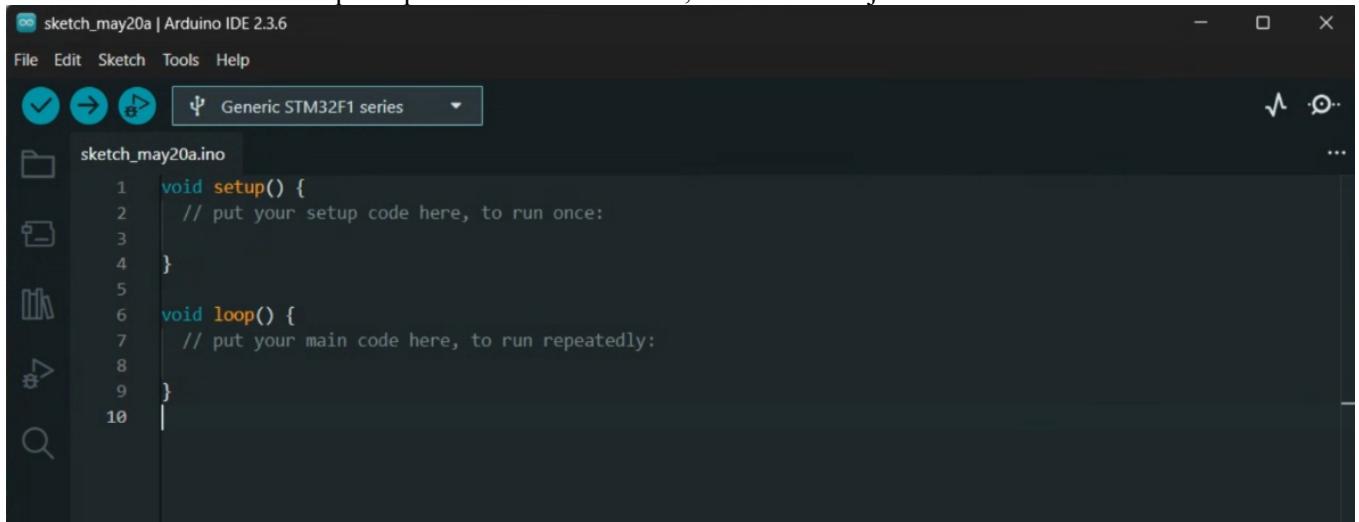


Fig. 4.1 - Fereastra principală dedicată scrierii codului

Codul Arduino este structurat în mod obișnuit în următoarele părți principale:

Funcția setup(): Aceasta este prima funcție care se execută în codul Arduino. Este folosită pentru a inițializa setările și configurațiile inițiale ale dispozitivului. În această secțiune, putem configura pinii de intrare/ieșire, inițializăm variabile, să setăm ratele de comunicare sau inițializăm module și senzori;

Funcția loop(): Această funcție este executată în buclă respectivă după ce funcția setup() a fost rulată o singură dată. Aici se află codul principal al programului și este locul în care se desfășoară majoritatea acțiunilor dispozitivului Arduino. Orice cod din interiorul funcției loop() va fi repetat în mod continuu, până când dispozitivul este oprit sau alimentarea este întreruptă;

Declarația de variabile: Variabilele pot fi de diferite tipuri, cum ar fi întregi (int), zecimale (float), caractere (char) sau adevărat/fals (boolean). De asemenea, pot fi declarate și constante pentru a stoca valori constante în timpul execuției programului:

- 'HIGH'/'LOW' – sunt utilizate pentru a reprezenta nivelurile de tensiune respectiv cu starea unui pin digital. 'HIGH' reprezintă nivelul de tensiune de 5V, care este considerat "1" logic sau "ON", 'LOW' reprezintă nivelul de tensiune 0V, care este considerat "0" logic sau "OFF".
- 'INPUT' / 'OUTPUT' – sunt utilizate pentru a seta pinii de intrare/ieșire.

În plus, în funcția `setup()` și `loop()` se pot folosi și alte instrucțiuni de control, cum ar fi instrucțiuni de decizie (`if, else, switch`), instrucțiuni de buclă (`for, while`), instrucțiuni de întrerupere (`interrupts`), etc., pentru a controla fluxul programului și a realiza diferite acțiuni în funcție de anumite condiții.

Functii personalizate: În codul Arduino, poți crea și crea funcții personalizate pentru a organiza mai bine codul și pentru a realiza sarcini speciale. Aceste funcții pot fi apelate din funcția `loop()` sau din alte funcții și pot include orice secvență de cod:

- `pinMode()` – stabilește rolul unui pin (intrare/ieșire);
- `digitalWrite()` – setează nivelul de tensiune al pinului digital;
- `digitalRead()` – preia valoarea intrării pinului digital;
- `analogRead()` – preia valoarea intrării analogice;
- `delay()` – oprește programul pentru timpul specificat în milisecunde dat ca parametru;

- O zonă de debugging, respective de comunicare a erorilor de compilare:

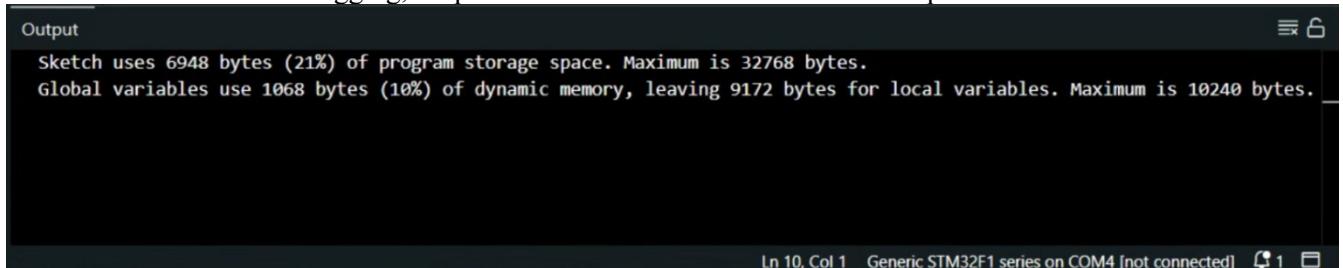


Fig. 4.2 - Zona dedicată debugging-ului

- Zona de instrumente, cu ajutorul cărora se compilează, încarcă codul și ajută la comunicarea cu microcontrollerul, prin intermediul porturilor seriale (ex. COM4):

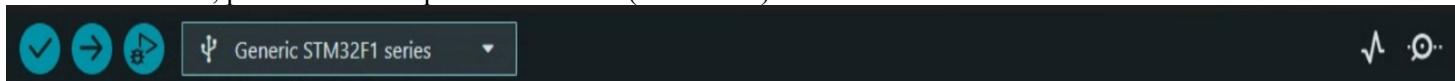


Fig. 4.3 - Zona dedicată instrumentelor

1. Verifică programul pentru erori;
2. Încarcă programul în placă de dezvoltare;
3. Lanseză modul de depanare;
4. Vizualizarea valorilor senzorilor;



5. Comunicarea prin portul serial cu calculatorul.

- O serie de unelte care ajută la plotarea și afișarea anumitor date (tensiuni citite de pe pini analogici), cum reprezintă o cantitate fizică, transformată într-o mărime electrică, prin intermediul unor senzori:

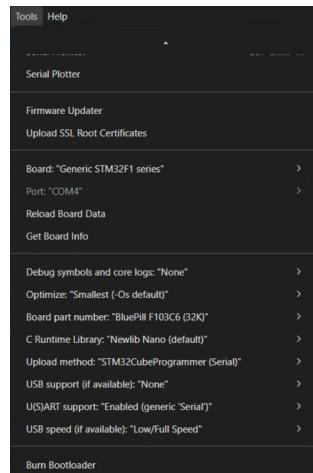


Fig. 4.4 - Zona dedicată uneltelor suplimentare

Proiectarea schemelor electrice:

Pentru realizarea circuitului aferent sistemului de urmărire a luminii, toate **schemele electrice** au fost proiectate utilizând software-ul **KiCad**, o platformă open-source dedicată proiectării PCB-urilor și schemelor electronice. KiCad a oferit un mediu intuitiv și complet pentru trasarea conexiunilor dintre componente precum **fotorezistori**, **rezistențe**, **servomotorul SG90**, **celula solară** și microcontroler. Utilizarea acestui instrument a permis o **documentare clară și precisă** a structurii circuitului, facilitând atât simularea logică, cât și trecerea ulterioară la faza de implementare practică pe breadboard sau PCB.

KiCad a permis structurarea proiectului în mod organizat prin utilizarea modulelor specifice:

- **Schematic Editor** – pentru desenarea schematicii electrice;
- **Symbol Editor** – pentru modificarea sau crearea simbolurilor personalizate ale componentelor;
- **Footprint Editor** – pentru asocierea simbolurilor cu amprentele fizice corespunzătoare de pe PCB;
- **Drawing Sheet Editor** – pentru maparea simbolurilor cu footprinturile;
- **PCB Editor** – pentru proiectarea layoutului plăcii de circuit imprimat;
- **Gerber Viewer** – pentru vizualizarea fișierelor finale de fabricație.

Construcția schematică:

În **Schematic Editor**, au fost plasate și conectate toate componentele necesare:

- **Fotorezistorii (LDR)** au fost integrați într-o rețea de divizori de tensiune cu rezistențe de $10\text{ k}\Omega$, pentru a crea semnale analogice proporționale cu intensitatea luminii.
- **Servomotorul SG90** a fost conectat la microcontroler prin pinul de semnal, iar alimentarea sa a fost prevăzută separat pentru a evita fluctuațiile.
- **Celula solară** a fost integrată ca sursă de alimentare alternativă, fiind protejată printr-un eventual circuit de reglare sau diode de blocare.

5. Implementare hardware

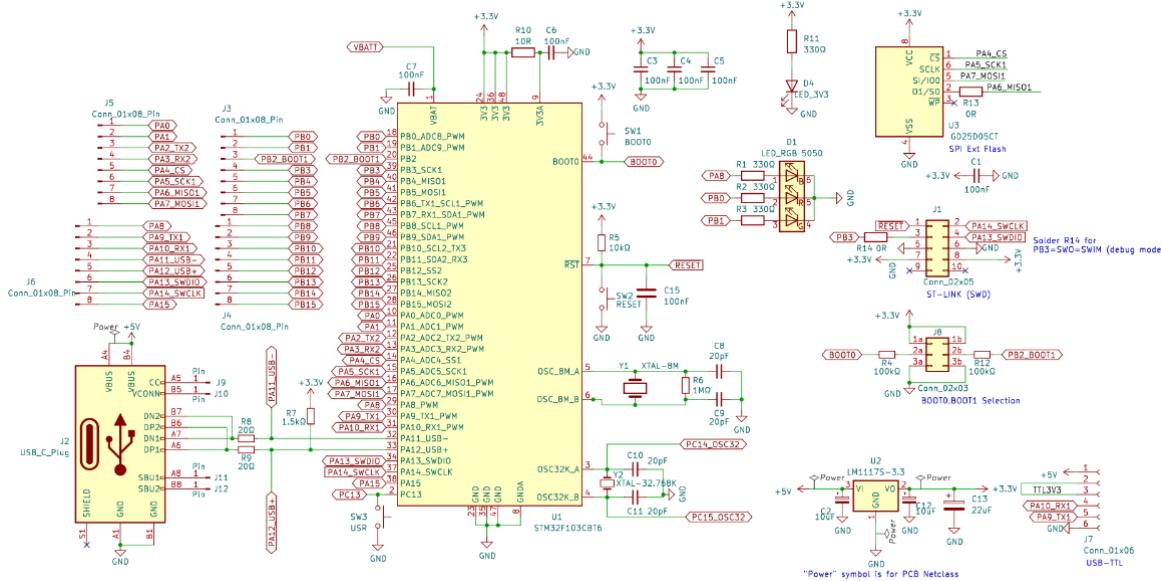


Fig. 5.1 – Schemă electrică a placii STM32

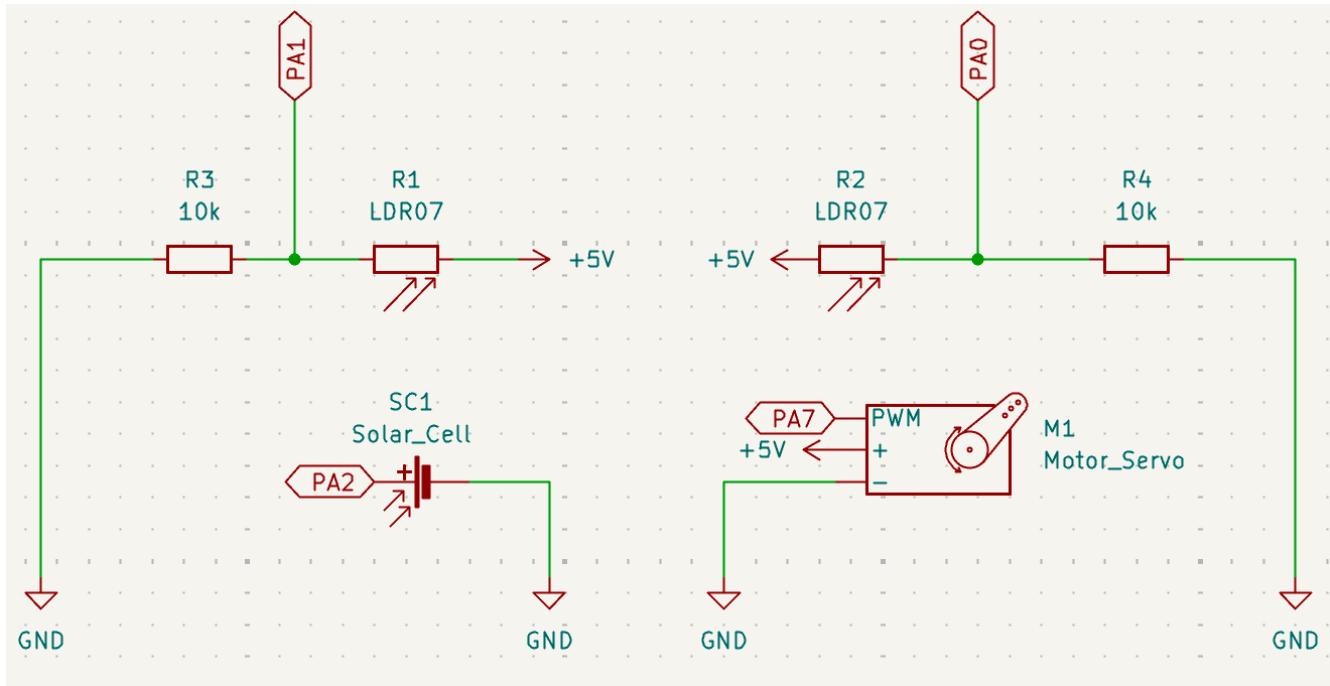


Fig. 5.2 – Schemă electrică a circuitului

Fotorezistorul este un element pasiv de circuit caruia îi variază rezistența în funcție de intensitatea luminii receptionate. În general, fotorezistentele sunt de putere mică și au rezistențe incepând de la câteva kΩ la lumina până la rezistențe de ordinul MΩ-lor în întuneric. Cea mai simplă aplicație este un circuit cu un divizor rezistiv format din fotorezistor și o altă rezistență de valori mari (10 kΩ). Prin variația rezistențelor la fotorezistor, variază și tensiunea de ieșire.

Schema de conectare este următoarea:

- Un capăt al **LDR-ului** se conectează la tensiunea de alimentare de **+5V**
- Celălalt capăt al LDR-ului se conectează la un capăt al **rezistenței fixe de 10 kΩ**
- Celălalt capăt al rezistenței se leagă la **GND (masă)**
- Din **nodul comun dintre LDR și rezistență** se extrage un fir, care se conectează la un **pin analogic** al microcontrolerului

Prin această configurație, la pinul analogic se obține o **tensiune variabilă în funcție de intensitatea luminii**. Când lumina crește, rezistența LDR-ului scade, iar tensiunea măsurată la nodul intermediu crește. Aceste variații sunt citite de microcontroler și transformate în valori numerice, permitând comparații între senzorii montați în diferite direcții.

Servomotorul **SG90** este un microservomotor foarte popular, utilizat frecvent în proiecte embedded datorită dimensiunilor compacte și controlului precis al poziției. Acesta dispune de **trei pini de conectare**, fiecare având un rol bine definit:

- **GND (fir maro)** – se conectează la **masa comună** a circuitului.
- **VCC (fir roșu)** – se conectează la o sursă de **5V**, tensiunea de alimentare a servomotorului.
- **Signal (fir portocaliu)** – primește semnalul de control de tip **PWM (Pulse Width Modulation)** de la microcontroler (ex: STM32 sau Arduino).

Semnalul PWM are o **frecvență de 50 Hz**, adică un ciclu la fiecare 20 ms. Poziția axului servomotorului este determinată de **durata impulsului (duty cycle)**, conform următoarei relații:

- Un impuls de **1 ms** poziționează axul la **0°** (extremitatea stângă).
- Un impuls de **1.5 ms** aduce axul la **90°** (poziția centrală).
- Un impuls de **2 ms** rotește axul la **180°** (extremitatea dreaptă).

Prin variația duratei impulsului în acest interval, se pot obține toate pozițiile intermediare între 0° și 180°.

O **celulă solară**, cunoscută și sub denumirea de **panou fotovoltaic**, este un dispozitiv electronic care convertește direct energia luminii solare în **energie electrică** prin efectul fotovoltaic. Este compusă din materiale semiconductoare (cel mai frecvent siliciu) care generează curent electric atunci când sunt expuse la lumină.

În cadrul unui sistem de **urmărire a luminii (solar tracker)**, celula solară poate avea două roluri:

a) Rol funcțional: sursă de alimentare

În unele aplicații autonome, mai ales în proiecte de tip **green energy** sau **IoT în zone izolate**, celula solară este utilizată pentru a **alimenta întregul circuit**, inclusiv microcontrolerul, senzorii și servomotorul. Energia generată de panoul solar este stocată într-un **acumulator (baterie Li-ion)** sau utilizată direct, în funcție de consum și condițiile de iluminare.

b) Rol de evaluare: optimizarea orientării

Celula solară poate fi folosită și pentru **măsurarea eficienței orientării**. De exemplu, se pot monta **două sau mai multe celule mici în diferite direcții**, iar sistemul poate alege orientarea optimă în funcție de ieșirea electrică maximă. Alternativ, o singură celulă solară mare poate fi orientată permanent de servomotor, astfel încât să primească **cea mai mare cantitate de lumină solară**.

Performanța unei celule solare este influențată de mai mulți factori:

- **Intensitatea luminii incidente**
- **Unghiul de incidență al razelor solare**
- **Temperatura mediului**
- **Curățenia suprafeței celulei**

Prin utilizarea unui sistem de urmărire a luminii, un panou solar poate genera **cu până la 25–35% mai multă energie** comparativ cu unul fix, deoarece este orientat permanent către sursa de lumină – adică Soarele.

Într-un sistem de urmărire solară (solar tracker), doi fotorezistori sunt montați de o parte și de alta a unui panou sau obstacol. Microcontrolerul compară valorile analogice generate de acești doi senzori. Dacă diferența de lumină este semnificativă, servomotorul SG90 este comandat să rotească panoul în direcția senzorului cu lumină mai intensă. Acest proces continuu permite sistemului să **urmărească sursa de lumină** pe tot parcursul zilei, optimizând expunerea la lumină (de exemplu, pentru panouri solare).

Combinația dintre **fotorezistori**, un **servomotor de precizie SG90** și o **celulă solară** oferă un sistem simplu, dar eficient de urmărire a luminii. Un astfel de sistem poate fi folosit atât în scopuri educaționale, cât și în aplicații practice precum panouri solare inteligente, sisteme de iluminat autonome sau dispozitive IoT alimentate solar.

6. Implementare software

- se ține butonul SW1 - BOOT0 apăsat
- se apasă scurt butonul RST (Reset) și se eliberează. În acest moment, procesorul pornește după reset și găsește BOOT0=1 (din cauza butonului) deci execuția softului se va face din "system memory" unde se află bootloaderul
- se eliberează butonul SW1 - BOOT0

Am rulat codul `stm32flash.exe -b 115200 -w test1.bin COM5` într-un Command Prompt pentru a testa funcționalitatea placii și a verifică dacă ledul tricolor se aprinde.

`stm32flash.exe`

Este un program folosit pentru a comunica cu microcontrolere STM32 prin port serial (USART bootloader), fără a folosi ST-Link sau alte programatoare

`-b 115200`

Setează **viteza de transfer serial** la 115200 bauds (comună și rapidă pentru bootloader)

`-w test1.bin`

Înseamnă "**write**" – scrie fișierul test1.bin în memoria flash a microcontrolerului

`COM5`

Este portul serial pe care este conectată placa ta STM32

Cod de testare servo-motor:

```
#include <Servo.h>

Servo myServo; // creează un obiect servo

void setup() {
    myServo.attach(PA0); // atașează servomotorul la pinul PA0 (sau orice alt pin PWM disponibil)
}

void loop() {
    myServo.write(0); // poziționează la 0 grade
    delay(1000); // așteaptă 1 secundă

    myServo.write(90); // poziționează la 90 de grade
    delay(1000);

    myServo.write(180); // poziționează la 180 de grade
    delay(1000);
}
```

Cod complet utilizat pentru functionarea intregului proiect:

```
#include <Servo.h>

// Pini
const int light1Pin = PA0;
const int light2Pin = PA1;
const int solarPin = PA2;
const int servoPin = PA7;

Servo servol;

int margin = 50;      // prag diferență lumină
int angle = 90;       // poziție virtuală între 0-180
int step = 1;          // pas la fiecare ciclu (echivalent mișcare)

unsigned long moveDuration = 100; // cât timp să ruleze motorul pe ciclu (ms)

void setup() {
    servol.attach(servoPin);
    servol.write(90); // stop
    Serial.begin(9600);
}

void loop() {
    int light1Reading = analogRead(light1Pin);
    int light2Reading = analogRead(light2Pin) - 120;
    int difference = light2Reading - light1Reading;

    // Citire panou solar
    int solarReading = analogRead(solarPin);
    float solarVoltage = solarReading * 5.0 / 1023.0;

    Serial.print("Sensor 1: ");
    Serial.print(light1Reading);
    Serial.print(" Sensor 2: ");
    Serial.print(light2Reading);
    Serial.print(" Voltage: ");
    Serial.print(solarVoltage);
    Serial.print("V Virtual angle: ");
    Serial.println(angle);

    // Rotație spre dreapta (crește unghiul)
    if (difference > margin && angle < 360) {
        servol.write(100); // rotire dreapta
        delay(moveDuration);
        servol.write(90); // stop
        angle += step;
    }
}
```

```

}

// Rotatie spre stanga (scade unghiul)
else if (difference < -margin && angle > 0) {
    servo1.write(80);           // rotire stanga
    delay(moveDuration);
    servo1.write(90);           // stop
    angle -= step;
}

// Daca diferența e mică => stai pe loc
else {
    servo1.write(90); // stop
}

delay(100); // pauza intre cicluri
}

```

Organograma codului:

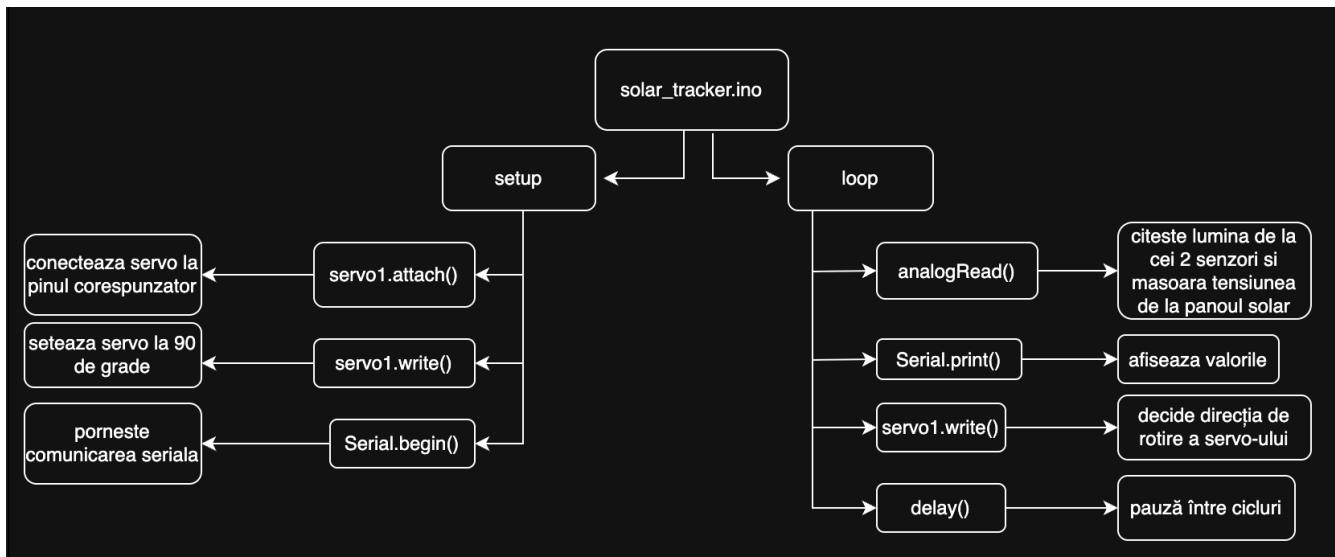


Fig. 6.1 – Diagrama bloc a codului

Explicatiile codului:

1. Declarații și inițializări:

- Include biblioteca pentru controlul servomotorului
- Definește pinii folosiți: 2 senzori de lumină (fotorezistori), un panou solar și un pin pentru servo
- Creează un obiect servo1 pentru a controla motorul
- margin: pragul minim de diferență între senzori pentru a considera că trebuie să se miște
- angle: poziția virtuală a servo-ului (nu este citită de motor)
- step: cu cât se modifică unghiul la fiecare mișcare
- moveDuration: timpul (ms) cât servo-ul se mișcă într-o direcție înainte să opreasă

2. setup() – Inițializare:

- Conectează servo-ul la pinul PA7
- Setează servo-ul în poziție neutră (90, adică stop pentru majoritatea servo-urilor continue)
- Pornește comunicarea serială pentru debugging

3. loop() – Cod care se execută continuu:

a) Citește valorile de la senzori:

- Citește lumina de la cei 2 senzori.
- La light2Reading se scade 120 pentru a calibra diferențele între senzori.
- difference: dacă e pozitiv, lumina e mai puternică pe senzorul 2; dacă e negativ, pe senzorul 1.

b) Măsoară tensiunea de la panoul solar:

- Măsoară tensiunea analogică și o convertește în volți (presupunem 5V ca referință ADC).

c) Afisează informațiile:

- Afisează valorile senzorilor, tensiunea panoului solar și poziția virtuală a servo-ului.

d) Decide direcția de rotire a servo-ului:

- Dacă lumina e semnificativ mai mare pe senzorul 2 și unghiul nu a depășit limita superioară atunci rotește servo-ul puțin spre dreapta (valori >90, ex. 100).
- Dacă lumina e semnificativ mai mare pe senzorul 1 și unghiul nu e sub zero: atunci rotește servo-ul puțin spre stânga (valori <90, ex. 80).
- Dacă lumina e aproape egală pe ambele părți, servo-ul se oprește.

4. Pauză între cicluri:

- Așteaptă 100ms între fiecare ciclu pentru a preveni oscilații rapide.

7. Concluzii

Proiectul realizat a avut ca obiectiv construirea unui sistem automat de urmărire a luminii, folosind o celulă fotovoltaică orientabilă, controlată de un servomotor și un microcontroler programabil. Sistemul utilizează doi senzori LDR pentru a detecta diferențele de intensitate luminoasă și ajustează automat poziția panoului în funcție de aceste valori, cu scopul de a simula comportamentul unui tracker solar. În implementarea sa, proiectul a combinat cunoștințe din domenii precum electronică aplicată, programare embedded, senzori, actuatori și proiectare de circuite.

Utilizarea mediului de dezvoltare Arduino IDE și a librăriei Servo.h a permis o programare intuitivă și eficientă, iar testarea s-a realizat cu succes prin monitorizarea valorilor în timp real, folosind Serial Monitor. De asemenea, proiectarea schematică electrică în KiCad a oferit o imagine clară și structurată a circuitului, facilitând o bună înțelegere a legăturilor dintre componente și susținând posibilitatea unei eventuale treceri la o versiune pe PCB.

Pe lângă componenta tehnică, proiectul a reprezentat o oportunitate excelentă de consolidare a abilităților de lucru în echipă, rezolvare a problemelor și documentare tehnică. În plus, a oferit un exemplu concret de aplicare a tehnologiei pentru optimizarea eficienței energetice, într-un context tot mai relevant legat de sustenabilitate și energie verde.

În concluzie, sistemul propus a demonstrat funcționalitatea unui mecanism de urmărire a luminii la scară redusă, fiind atât un instrument educațional util, cât și o bază solidă pentru dezvoltări viitoare, precum adăugarea unei a două axe de mișcare, integrarea cu senzori de mediu sau colectarea de date pentru analiză și optimizare în timp real.

8. Bibliografie

P2-STM32-Intro - Documente Google

https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json

<https://www.arduino.cc/>

<https://www.kicad.org/>

<https://www.youtube.com/watch?v=nkPoRdrsRHE>

<https://docs.arduino.cc/libraries/servo/>

https://www.sciencebuddies.org/science-fair-projects/project-ideas/Energy_p045/energy-power/solar-tracker