Solar tracker

Autor: Iustina-Andreea Cărămidă
Grupa: 332CA
Email: iustina.caramida@stud.acs.upb.ro

Introducere

În lumea modernă, energia solară devine tot mai importantă pentru a satisface nevoile noastre de energie într-un mod sustenabil și ecologic. Cu toate acestea, pentru a maximiza eficiența panourilor solare, acestea trebuie să fie aliniate corect către sursa lor de energie primară: Soarele.

Proiectul se concentrează pe automatizarea acestei alinieri, scopul principal fiind crearea unui sistem care să monitorizeze poziția Soarelui și să ajusteze orientarea panoului solar în consecință, astfel încât să maximizeze captarea energiei solare.

Însă, proiectul are o abordare inovatoare: nu ne limităm doar la maximizarea captării de energie solară în timpul zilei, ci dorim să pregătim panoul solar pentru o nouă zi încă de la apusul Soarelui. Astfel, panoul solar se va poziționa automat spre direcția de răsărit, pregătindu-se astfel pentru a începe o nouă zi de captare a energiei solare încă de la primele raze ale dimineții.

În cele ce urmează, vom detalia componentele necesare, conexiunile și codul Arduino utilizat pentru a realiza acest sistem inovator și eficient energetic.

Descriere generală

Dispozitivul va utiliza un servomotor pentru a ajusta alinierea panourilor solare către direcția de unde provine cea mai intensă lumină solară, detectată de cei doi fotorezistori, puși în direcții opuse pentru a capta un mediu mai larg. Microcontroller-ul va coordona acțiunile servomotorului pentru a evalua mediul înconjurător. Informațiile furnizate de fotorezistori vor fi analizate și utilizate pentru a determina poziția optimă față de sursa de lumină, după care dispozitivul va ajusta orientarea panourilor solare în acea direcție.

Mai jos puteți găsi o schemă high-level a proiectului:

Hardware Design

Componentele necesare sunt:

Obiect	Cantitate
Arduino UNO board	1
Panou solar	1
SG90 servo motor	1
senzori LDR	2
rezistență 10k	3
rezistență 220	1
LED	1
buton	1
fire	multe
poliester	mult

Schema electrică este următoarea:

Design-ul montări este umrătorul:

Software Design

Link către repo-ul de Github cu codul sursă: https://github.com/iuniod/Solar-Tracker

Elementul de noutate al acestui proiect constă în integrarea controlului precis al unui panou solar utilizând senzori LDR pentru a maximiza eficiența energetică. Utilizarea întreruperilor pentru a realiza temporizări precise și gestionarea stării panoului printr-un buton de control reprezintă o abordare inovatoare pentru un sistem de urmărire a soarelui simplu și eficient.

În cadrul proiectului au fost utilizate următoarele funcționalități:

- **GPIO (General Purpose Input/Output)**: Utilizat pentru a controla pinul LED-ului și a citi starea butonului, esențial pentru interacțiunea hardware-software.
- **UART (Universal Asynchronous Receiver-Transmitter)**: Utilizat pentru comunicarea serială, permite monitorizarea și debug-ul sistemului prin afișarea mesajelor în Serial Monitor.
- **Interrupții**: Folosite pentru a implementa o temporizare precisă folosind Timer2, necesară pentru funcțiile de întârziere fără a bloca execuția.
- **Timere**: Timer2 a fost utilizat pentru a genera întreruperi la intervale precise de timp, esențiale pentru temporizările exacte în funcția delay_ms().
- **ADC (Analog-to-Digital Converter)**: Folosit pentru a citi valorile analogice de la senzorii LDR și a le converti în valori digitale utilizabile în algoritmul de control al panoului.

Calibrarea senzorilor LDR a fost realizată prin determinarea valorilor minime și maxime de lumină detectate în diferite condiții de iluminare. Pragul de eroare (error) și valoarea epsilon (eps) au fost ajustate empiric pentru a asigura că panoul se poziționează corect în funcție de diferențele detectate între cei doi senzori.

Optimizările au fost realizate în următoarele moduri:

- **Utilizarea întreruperilor și a Timer2**: Acest lucru a permis implementarea unei funcții de întârziere precise (delay_ms()) fără a bloca execuția altor părți ale codului, optimizând astfel eficiența și responsivitatea sistemului.

Mediul de Dezvoltare

Codul Arduino este dezvoltat utilizând mediul Arduino IDE (Integrated Development Environment), un mediu software folosit pentru a scrie, compila și încărca codul pe plăcile de dezvoltare Arduino. Arduino IDE oferă o interfață simplă și intuitivă, suportând limbajul de programare C/C++ cu adăugiri specifice platformei Arduino. IDE-ul include un editor de cod, un monitor serial pentru debugging și multe alte instrumente utile pentru dezvoltarea de proiecte embedded.

Librării și Surse Terțe

În codul prezentat sunt utilizate două librării esențiale:

- **Servo.h**: Aceasta este o librărie standard în Arduino pentru controlul servomotoarelor. Permite utilizatorilor să controleze poziția unui servomotor folosind funcția write().
- **avr/interrupt.h**: Aceasta este o librărie specifică microcontrolerelor AVR utilizate de multe plăci Arduino. Permite gestionarea întreruperilor, esențiale pentru implementarea unor funcții precum temporizări precise și reacții rapide la evenimente externe.

Organizarea Codului

Codul este organizat logic în mai multe secțiuni:

Importul librăriilor

Importă librăriile necesare pentru funcționalitatea codului.

```
// ------------------- Import libraries --------------------
// Include the servo motor library
#include <Servo.h>
// Include the interruption library
#include <avr/interrupt.h>
```

Definirea pinilor și a erorilor

Definirea pinilor pentru senzori și alte componente hardware, precum și valorile de eroare utilizate în algoritmi.

```
// ---------------- Define pins and errors -----------------
// Define the LDR sensor pins
#define LDR1 A0
#define LDR2 A1
// Set epsilon value
#define eps 10
// Set error code
#define error 1000
// Starting point of the servo motor
int Spoint =  0;
// Create an object for the servo motor
Servo servo;
// Select the pin for the LED
const int ledPin = 12;
// Select the pin for the button
const int buttonPin = 13;
```

Variabile globale

Definirea variabilelor globale necesare pentru stocarea stării sistemului și a temporizărilor.

```
// ---------------- Define global variables ----------------
// Global variable to count milliseconds
volatile unsigned int timer2_millis = 0;
// Global variable to count the time since last position has moved
volatile unsigned int lastPositionChange = 0;
// Global variable to check the button
volatile bool ifPressed = false;
```

Funcții utile

Implementarea funcțiilor esențiale pentru funcționarea programului, cum ar fi funcția de întrerupere pentru Timer2, funcția personalizată de întârziere, funcția de mișcare a servomotorului, mesaje pentru utilizatori și funcția de iluminare a LED-ului.

Funcția setup()

Configurarea inițială a hardware-ului, inclusiv atașarea servomotorului, configurarea pinilor și

inițializarea temporizărilor.

Funcția loop()

Bucla principală care rulează continuu și care gestionează citirea datelor de la senzori, verificarea stării butonului, mișcarea servomotorului, afișarea mesajelor pentru utilizatori și controlul LED-ului.

Algoritmi și Structuri Implementate

Codul utilizează mai mulți algoritmi și structuri pentru a controla mișcarea unui panou solar:

Controlul Servomotorului

Utilizând valori de la doi senzori LDR (Light Dependent Resistor), codul calculează diferența dintre citirile acestora pentru a ajusta poziția servomotorului. Algoritmul verifică dacă diferența este în limite acceptabile (epsilon), și ajustează poziția panoului solar în consecință.

Gestionarea întreruperilor

Utilizarea întreruperilor prin Timer2 pentru a crea o funcție de întârziere precisă. Acest lucru este esențial pentru temporizări precise fără a bloca execuția altor părți ale codului.

Funcționalitate pe bază de stare

Codul utilizează o variabilă globală `ifPressed` pentru a determina dacă panoul solar trebuie să se miște sau să rămână într-o poziție fixă, în funcție de starea butonului.

Mesaje informative

Pe baza poziției curente a panoului solar, sunt afișate mesaje informative pentru utilizatori, cum ar fi „Good Morning sunshine!” sau „Good Afternoon!”.

Controlul LED-ului

LED-ul este controlat în funcție de timpul scurs de la ultima mișcare a panoului solar, indicând astfel dacă panoul a găsit o poziție bună pentru a se încărca.

Aceste structuri și algoritmi combină citirea senzorilor, controlul precis al hardware-ului și feedback-ul pentru utilizator într-un mod coerent și eficient pentru a controla un panou solar.

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună .

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** şi **Resurse Hardware**.

Export to PDF